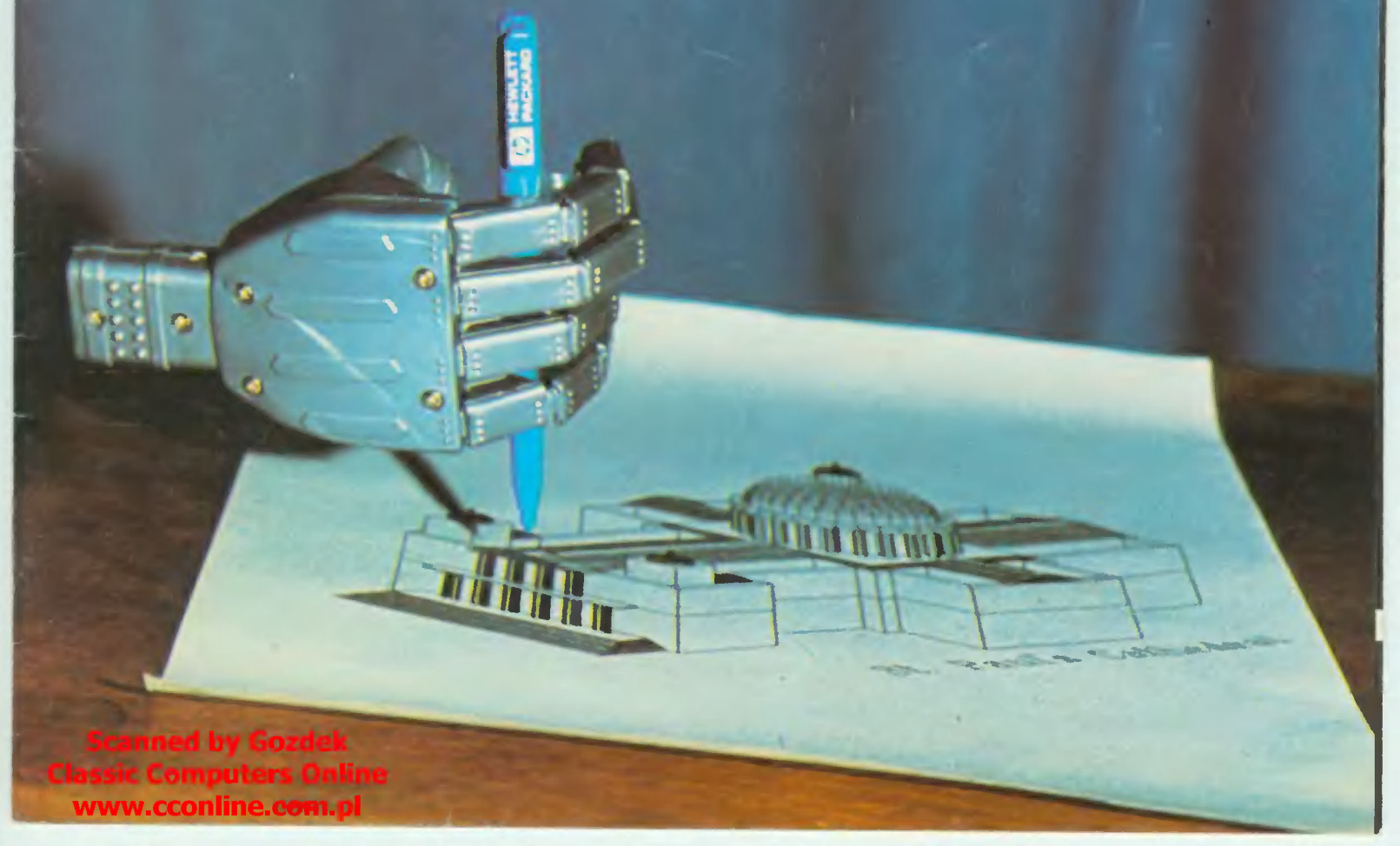
mikroklan

Magazyn informatyczny ISSN 0860-1941 SIGMA cena zł.395 2/88

CAD



Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

MICROSYSTEM

BIURO PROJEKTÓW I
ZASTOSOWAŃ INFORMATYCZNYCH

Microsystem Sp. z o.o. j.g.u. od 1985

KOMPLEKSOWY, WIELODOSTĘPNY, SIECIOWY
ZAUTOMATYZOWANY SYSTEM ZARZĄDZANIA

Do 30 stanowisk: IBM PC XT/AT i terminale

Wspólna baza danych

Sprawdzone, wdrażane w całym kraju podsystemy:

- Finansowo-Księgowy
- Zatrudnienie i Płace
- Gospodarka Materiałowa
- Informowanie Kierownictwa

SPRZĘT:

Komputery klasy IBM PC XT/AT, Amstrad/Schneider, drukarki, plotery, streamery

SZKOLENIA:

budowa, obsługa, zastosowanie komputerów

BPiZSI Microsystem

Centrala:

✉ 80-360 Gdańsk ul. Krzywoustego 44

Tx 051-2259 mics pl

☎ 52-34-49

Dział usług:

✉ 80-309 Gdańsk ul. Bażyńskiego 1

Tx 051-2749 hws pl

☎ 52-49-32

mikroklan 2(14) luty 1988

P-CAD 1
Więcej cali... 4
Mikrokram 5, 12, 25, 29
Żywa tablica 6
AutoCAD 2.6 7
Analyser 9
MS-DOS od środka 10
Zmiana LPT 1 na LPT 2 13
Wywołania z Turbo-Pascala 14
Klawiatury 15
Programowanie sterowników w Pascalu 19
Forth – niedoceniony 21
Sinus, Cosinus, Tanges 23
Turbo C 24
Workstations 26
Dyski optyczne 27
Elektoniczna poczta 28
Prosty generator liczb pseudolosowych 29
Trzeci wymiar 30
Elektroniczne deski kreślarskie 32

Redaktorem bloku o DPT w styczniowym numerze mikroklanu jest p. Krzysztof Wiśniewski, a autorem wszystkich zdjęć p. Aleksander Kuplicz — przepraszamy za pominięcie ich nazwisk (red).

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Komputer osobisty nie jest wymarzonym narzędziem dla naukowca czy inżyniera. Ograniczenia sprzętowe powodują, że dość często bardziej racjonalne jest stosowanie dużego systemu komputerowego, a czasem nawet wygodniejsze okazują się ... metody konwencjonalne. Są jednak prace naukowo-inżynierskie, w których komputer osobisty jest użyteczny. W artykułach o programach PCAD, MathCAD i nowej wersji AutoCAD-u opisane zostały przykłady zastosowań tego typu. Słowo użyteczny nie oznacza jednak optymalny. Wykorzystanie AutoCAD-u i PC AT do projektowania BMW pozwoliłoby konkurencyjnym firmom spać spokojnie. Złożone prace wymagają sprzętu o większych możliwościach - jest o tym mowa w artykule Workstations.

Dobór sprzętu, wyposażenia dodatkowego i właściwego oprogramowania przesądza o wyniku prac. Np. przygotowanie wydruku rysunku umieszczonego na okładce zajęło komputerowi PC 1512 HD20 ponad 7 godzin (!).

Czy stosowanie komputera tego typu do prac projektowych ma więc sens?

W Polsce dobór właściwego sprzętu przestaje być sprawą łatwą, gdy tylko obszar zastosowań wykracza poza rutynowe prace biurowe. Firmy zajmujące się sprzedażą i doradztwem mają fatalny nawyk doradzania rozwiązań dopasowanych nie tyle do potrzeb przyszłego użytkownika, ile do stanu swojego magazynu. Natomiast firmy spod hasła u nas najtaniej oferują ... możliwie kosztowną konfigurację sprzętu, bo profesjonalizm musi przecież kosztować.

Trudno dociec dlaczego sytuacja taka tolerowana jest przez nabywców - przy zrównoważonym rynku klient ma prawo wymagać nie tylko odpowiedniego sprzętu, dokumentacji i oprogramowania, ale i przeszkolenia. Szkolenie nie musi odbywać się na sprzęcie zakupionym - przeciwnie może stanowić podstawę do dalszych decyzji.

Spośród krajów RWPG dysponujemy chyba największą liczbą komputerów osobistych. Wiele z nich trafiło do pracowni konstruktorów i naukowców. Jednak rezultaty mamy zbliżone do państw, w których komputer osobisty jest rzadkością. Jest to wynik nieprofesjonalnego podejścia do profesjonalnych prac. Sprzęt jest kupowany ad hoc (za tzw. gorące pieniądze pod koniec roku), a oprogramowanie pozyskiwane od kolegi. Oczywiście bez instrukcji, bo mimo wielomilionowego wydatku na sprzęt najtrudniejszą do sforsowania, barierą jest zrobienie kserokopii.

O wyposażeniu każdego stanowiska w prasę komputerową nie ma co marzyć - przecież egzemplarz jest w bibliotece.

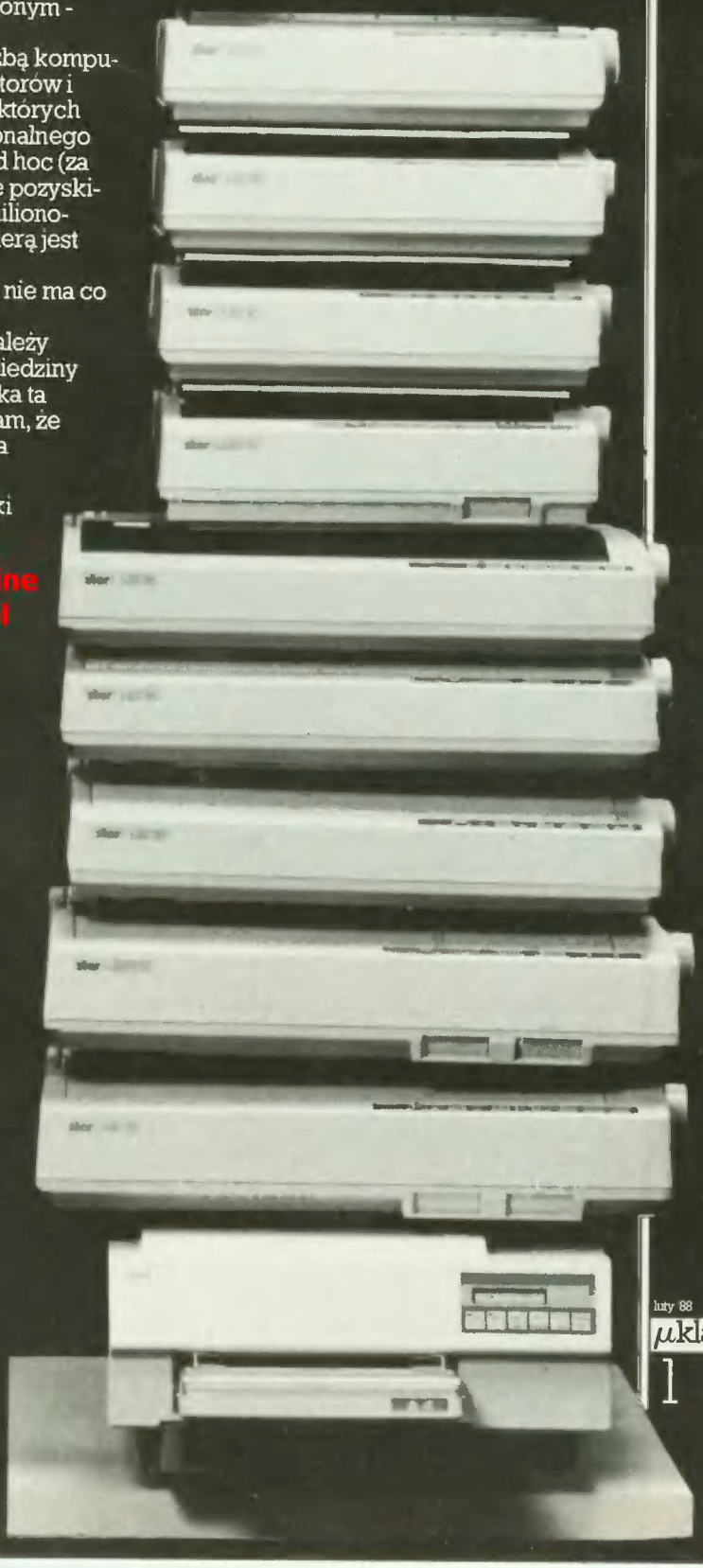
Materiały, które publikujemy w tym numerze **mikro**klanu należy traktować jako zasygnalizowanie niektórych zagadnień z dziedziny komputerowego wspomagania prac inżynierskich. Tematyka ta będzie często gościć na naszych łamach, gdyż wydaje się nam, że właśnie od rozwoju zastosowań CAD zależy praktycznie cała gospodarka.

Andrzej J. Piotrowski

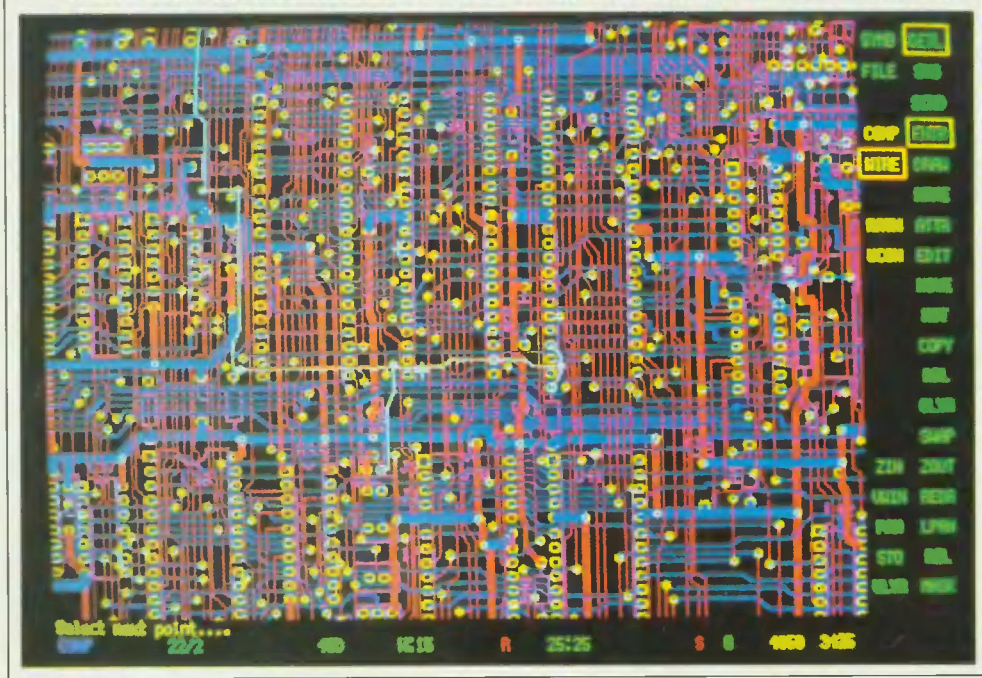
Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Redaguje zespół:

Redaktor naczelny – Andrzej J. Piotrowski
sekretarz redakcji – Maria Pawlak
z-ca sekr. red. Alicja Wancerz-Gluza.
redaktorzy: Tomasz Dec, Krzysztof Drozdowski, Roland Hojczak
Krzysztof Kamiński, Maciej A. Markowski,
Sławomir Piotrowski, Leszek B. Wroński, Krzysztof Rzymkowski,
Antoni Szendzielorz, Romuald Szuniewicz.
Stali współpracownicy: Zbigniew Pojmański, Andrzej Suwała,
Janusz Wrześniak.
Komputerowe przygotowanie tekstów: Metrum – Krystyna W. Piotrowska
Redaktor techniczny – Małgorzata Szumna
Zdjęcie na okładkach – Jan Stolarczyk
Adres redakcji: ul. Świętokrzyska 14a, PL 00-950 Warszawa, skrytka 1004.
Telefony: 26-20-27, 26-74-61 w. 473, 26-54-31 w. 376
Zlecenia na zamieszczenie ogłoszeń i reklam przyjmuje Dział Ogłoszeń i Reklamy Wydawnictwa NOT-SIGMA ul. Świętojerska 5/7, 00-236 Warszawa tel. 31-93-65, 31-22-21, wew 291, 196 oraz akwizytorzy:
Zbigniew Rutkowski tel. 33-62-77
Anna Moniuszko tel. 10-69-85
Wydawca: Wydawnictwo Czasopism i Książek Technicznych SIGMA.
Przedsiębiorstwo Naczelnej Organizacji Technicznej.
Skład: WZKart (Linotronic 300) – Małgorzata Maliszewska, Piotr Kurzec.
Stałe otrzymywanie magazynu informatycznego **mikro**klan można sobie zapewnić wysyłając zamówienie bezpośrednio do Zakładu Kolportażu Wydawnictwa NOT-SIGMA ul. Bartycka 20, 00-950 Warszawa, skrytka 1004.
Zamówienia można składać na wybrane numery oraz okresy kwartalne, półroczne i roczne w ramach danego roku kalendarzowego. Kontynuacja otrzymywania czasopisma w roku następnym wymaga ponownego złożenia zamówienia. Można również zamawiać egzemplarze archiwalne. Zamawiający nie dokonują przedpłat.
Instytucje podają w zamówieniu warunki płatności. Odbiorcy indywidualni otrzymują zamówione egzemplarze za zaliczeniem pocztowym równym cenie czasopisma (porto opłaca nadawca).
Informacji udziela Dział Handlowy.
Tel. W-wa 40-00-21 w. 207, 238 lub 40-37-31.
Druk: Bohmann Druck und Verlag GmbH co. KG Wien, Austria.
Druk na zlecenie Ars Polona.
Nakład 100 000 egz. K-65 ISNN 0860-1941 Cena 395zł.



P-CAD



System P-CAD jest przeznaczony do rysowania schematów ideowych, symulacji pracy układów cyfrowych oraz projektowania obwodów drukowanych. Umożliwia on wykonywanie pełnej dokumentacji urządzeń elektronicznych i ułatwia jej modyfikację przy wykorzystaniu prowadzonego archiwum.

System P-CAD może być stosowany na komputerach typu IBM PC, wyposażonych w 640 KB pamięci, jeden napęd dysków elastycznych 360 KB, dysk sztywny 20 MB, kartę P-CAD oraz monitor kolorowy. Komunikację z systemem umożliwia mysz lub klawiatura.

W skład pakietu P-CAD wchodzą programy: PC-CAPS, PC-CARDS, PC-LOGS, PC-NODES, PC-LINK, PC-PACK, PC-FORM, PC-PLOTS, PC-PRINT, PC-HISTO, PC-ROUTE, PC-DRC/NLC, PC-BACK, PREPACK, PRESIM, POSTSIM, PC-PHOTO. Programy PC-CAPS, PC-CARDS i PC-LOGS, wspomagane odpowiednimi zestawami pozostałych, tworzą podstawowe bloki funkcjonalne.

System P-CAD zawiera bogatą bibliotekę symboli w standardzie ANSI oraz IEEE. W wielu przypadkach konieczne jest jednak tworzenie własnych symboli. Wykorzystywane są wówczas polecenia rysowania figur geometrycznych oraz wprowadzania kompletnej informacji o wyprowadzeniach elektrycznych elementu, tj. wartości współrzędnych położenia, nazwie i funkcjach (we, wy, we-wy). Oddzielnym poleceniem wprowadzane są informacje wykorzystywane w procesie automatycznego projektowania płytek drukowanych. Dotyczą one oznaczenia elementu, liczby układów cyfrowych lub bramek w obwodzie i numerów wyprowadzeń odpowiadających danej bramce. Dodatkowe informacje dotyczące symbolu elementu, np. nazwa producenta lub technologia wykonania, są wprowadzane jako jego atrybuty. Wspomniane dwa rodzaje zestawów poleceń umożliwiają hierarchiczną budowę schematów. Powtarzające się fragmenty złożonych schematów, zbudowane z symboli bibliotecznych, mogą być definiowane jako nowe symbole (bloki), z których tworzony jest z kolei schemat wyższego stopnia. System umożliwia stosowanie do piętnastu stopni hierarchii. Innym sposobem tworzenia złożonych schematów jest budowanie jego fragmentów na oddzielnych rysunkach (warstwach). Każdej warstwie nadawany jest kolejny numer. Proces tworzenia schematu składa się z pobierania symboli z biblioteki i realizacji połączeń. Zbiór połączonych punktów o jednakowym potencjale tworzy sieć. Każda sieć jest identyfikowana przez widoczną nazwę lub numer nadawany przez program. Program minimalizuje liczbę błędnych połączeń w trakcie ich realizacji, tj. uniemożliwia połączenie sieci o różnych nazwach, a sieci nie nazwane są łączone dopiero po powtórnym zaakceptowaniu przez projektanta.

Program PC-CAPS umożliwia rysowanie schematów ideowych i tworzenie opisującej je bazy danych. Wprowadzone dane służą do wykonania rysunku na ploterze lub drukarce oraz utworzenia listy połączeń elementów. W dalszym etapie lista ta jest wprowadzana do bazy wejściowej dla programu PC-CARDS. W programie stosowana jest warstwowa struktura informacji graficznej. Dla każdej warstwy zostają zdefiniowane atrybuty: kolor, odpowiadający mu numer pisaka w ploterze oraz aktywność, czyli możliwość wprowadzenia na nią danych. Program umożliwia stosowanie do 54 warstw. Zredagowanie zbioru sterującego konkretnym urządzeniem wyjściowym, na którym ma być wykonana dokumentacja rysunkowa wymaga uprzedniej selekcji warstw.

Informacje graficzne wprowadzane są w prostokątnym układzie współrzędnych z siatką, dla której odległości między węzłami są zdefiniowane jako wielokrotności 0,01 cala. Maksymalne wymiary rysunku wykonanego w skali 1:1 wynoszą 600x600 cali. Do drukowania lub rysowania (programy PC-PLOT, PC-PRINT) przyjmuje się podziałkę zmniejszającą, ograniczoną jedynie czytelnością rysunku. Wprowadzany schemat lub jego fragment, zestaw poleceń, linie komunikatów i statusu tworzą pełny obraz na ekranie. Polecenia priorytetowe i polecenia drugiego stopnia są wybierane i uaktywniane za pomocą kursora. Istniejące zestawy poleceń odpowiadają dwom rodzajom pracy: tworzeniu nowych symboli (SYMB) oraz rysowaniu schematów (DETL). W obu zestawach zawarte są między innymi polecenia wprowadzania informacji graficznych, modyfikowania istniejącego rysunku, sterowania obrazem (zbliżenia, oddalenia, przesunięcia w inny obszar).

Po utworzeniu kompletnego schematu ideowego powstaje zbiór przyporządkowujący użytym w schemacie symbolom odpowiednie obudowy. Będą one pobierane przez program PC-PACK z biblioteki, tworząc wraz z listą połączeń wejściową bazę danych dla następnego etapu - projektowania obwodu drukowanego.

Programy PC-NODES i PC-FORM umożliwiają wzbogacenie dokumentacji o listy połączeń (tworzone pod kątem elementów lub sieci) oraz o listę materiałową.

Blok funkcjonalny systemu, służący do projektowania obwodów drukowanych, zawiera programy: PC-CARDS (interaktywny edytor), PC-HISTO (do szacowania gęstości upakowania obwodu drukowanego i wspomagania optymalnego rozmieszczania elementów na pakiecie), PC-ROUTE (do automatycznego prowadzenia ścieżek) oraz PC-PHOTO (tworzący zbiory sterujące fotoploterem).

Zasady obsługi programu PC-CARDS są podobne do stosowanych w PC-CAPS. Inicjalizacja, wybór opcji i konfiguracja ekranu przeprowadzone są w identyczny sposób.

Polecenia, tak jak w PC-CAPS, można podzielić na polecenia wprowadzania informacji graficznych, edycji i sterowania obrazem. Dodatkowe polecenia umożliwiają kontrolę nad miejscami zapewniającymi elektryczne połączenie między fizycznymi warstwami obwodu drukowanego (przepusty metalizowane). Istotną pomoc stanowią linie obrazujące bezpośrednie elektryczne połączenie między wyprowadzeniami elementu. Dodakowo PC-CARDS zawiera polecenia, którymi wprowadzane są informacje wykorzystywane w zbiorze sterującym fotoploterem i dotyczące numerów przesłon. Ich reprezentacjami graficznymi są punkty o stałych wymiarach, bez względu na numer stosowanej przesłony.

W PC-CARDS stosowana jest taka sama struktura warstwowa przy definiowaniu nowych elementów oraz przy tworzeniu całego obwodu drukowanego. Każda informacja dotycząca pól lutowniczych, ścieżek, napisów na pakiecie itp. jest wprowadzana na odpowiednią warstwę. Warstwy informacji graficznych, przeznaczonych do wyprowadzenia na ploter lub drukarkę, mają swoje odpowiedniki w warstwach zawierających informacje dla fotoplotera.

Pakiet P-CAD zawiera bogatą bibliotekę elementów, co znacznie ułatwia projektowanie obwodu drukowanego przy użyciu programu PC-CARDS. Użytkownik ma też możliwość tworzenia własnej biblioteki elementów. Wymaga to zdefiniowania parametrów elementu w sposób analogiczny, jak dla elementów zawartych w bibliotece, tj. typów wyprowadzeń, ich równoważności oraz informacji dotyczących bramek zawartych w elemencie.

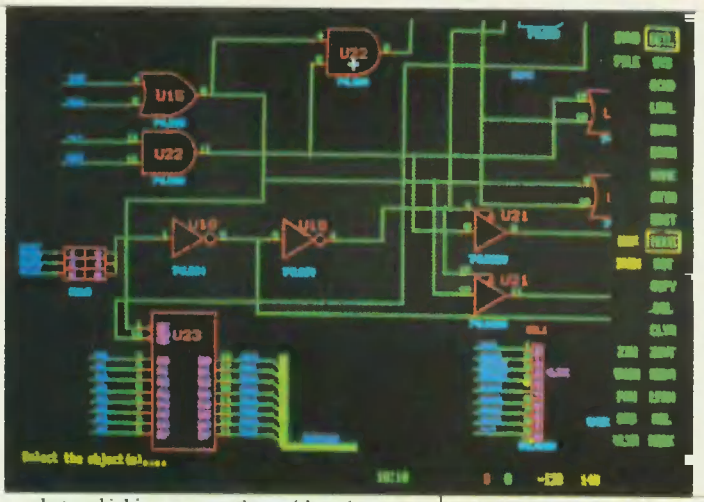
Typ wyprowadzenia określa wielkość pola lutowniczego, numer przysłony w fotoploterze, średnicę otworu itp. Równoważność elementów wykorzystywana jest do zamiany wyprowadzeń między sieciami. Informacje dotyczące bramek zawartych w elementach są wymagane przez program transformujący (PC-PACK). Zezwalają także na zamianę bramek w elemencie lub między elementami.

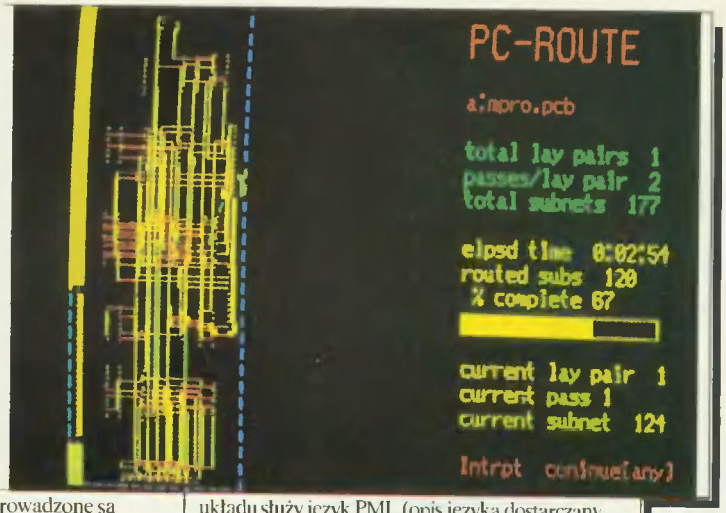
Rozmieszczenie połączeń elektrycznych na projektowanej płytce (trasowanie) może być wykonywane z wykorzystaniem informacji o połączeniach, zawartych w schemacie ideowym (utworzonym za pomocą programu PC-CAPS) lub z ich pominięciem - na podstawie listy połączeń. Trasowanie może być realizowane automatycznie lub przez operatora za pomocą digitizera (przy wykorzystaniu myszy). W pracy konwersacyjnej, system kontroluje na bieżąco poprawność połączeń sieci oraz zgodność połączeń z wejściową listą połączeń. Po uzyskaniu kompletnej sieci, na ekranie następuje wygaszenie odpowiedniej linii bezpośredniego połączenia.

Program ten nie przewiduje jednak automatycznej kontroli wymaganych parametrów obwodu drukowanego (minimalna odległość między ścieżkami, przejściami oraz minimalne szerokości ścieżek). Do sprawdzenia parametrów służy projektantowi program PC-DRC/NLC. Wyniki testowania podawane są w postaci tekstu (raport) lub w formie graficznej - na dodatkowych, utworzonych przez PC-DRC/NLC warstwah projektu.

W trakcie prowadzenia ścieżek i rysowania linii, określane są ich atrybuty takie jak: nazwa, szerokość (0... 0,250), rodzaj linii (ciągła, przerywana)

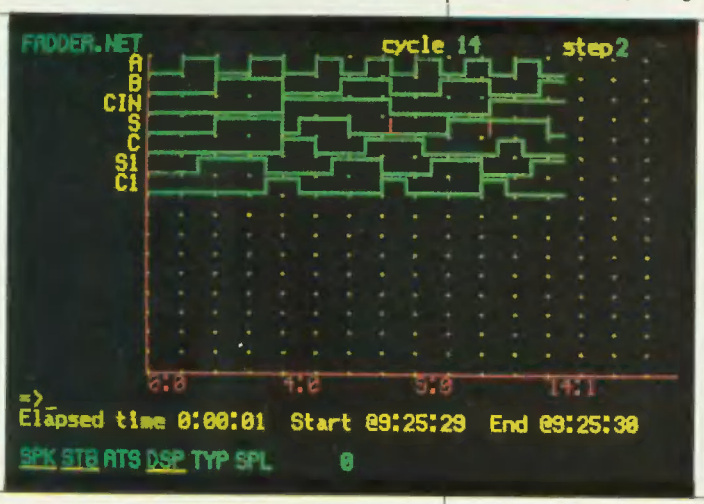
Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

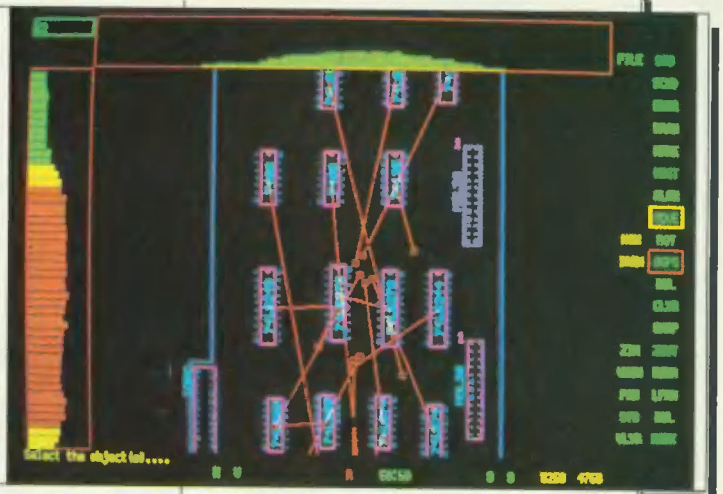




oraz kąt pod jakim są prowadzone (dowolny, 45° lub 90°).
Program zawiera zestaw poleceń służących do rysowania, edycji, obrotu, przesuwania, kopiowania oraz wprowadzania elementów, ścieżek i przejść między warstwami obwodu. Polecenia dotyczą pojedynczych elementów lub ich grup. Polecenia sterowania obrazem dają możliwość dziesięciokrotnego zmniejszania lub powiększania obrazu, przesunięcia w inny obszar, zapamiętania i odtworzenia 10 obrazów oraz odpowiedniego powiększania wybranego obszaru. Obwód drukowany o maksymalnych wymiarach 60x60 cali może zawierać 500 elementów, 1000 sieci i łącznie 8000 wyprowadzeń w układach. Wszystkie informacje graficzne wprowadzane są w siatce, której raster określony jest jako wielokrotność 0,001 cala.
Do automatycznego wprowadzania ścieżek wykorzystywane są programy PC-HISTO i PC-ROUTE. Program PC-HISTO dostarcza niezbędnych informacji (histogram i wektory połączeń), umożliwiając optymalne rozmieszczenie elementów na płytce.
Histogram przedstawia stopień wykorzystania kanałów trasowania, tzn. stosunek liczby bezpośrednich połączeń, przecinających odpowiednio poziome i pionowe linie siatki trasowania, do wszystkich teoretycznie możliwych kanałów. Korekcja histogramu i wektorów następuje automatycznie ze zmianą położenia elementów. Po zaakceptowaniu rozmieszczenia elementów i uruchomieniu programu PC-ROUTE, następuje automatyczne trasowanie. Istnieje możliwość wyłączenia określonych przez projektanta obszarów. Możliwe jest również wstępne wprowadzenie sieci, np. masy i zasilania, które będą zaakceptowane przez program.
Program PC-ROUTE umożliwia określenie różnych strategii prowadzenia ścieżek. Opracowanie strategii wymaga podania parametrów trasowania oraz parametrów ścieżek i pól lutowniczych. Zasadnicze parametry określające sposób trasowania to: siatka podstawowa, typ połączeń poszczególnych segmentów sieci, porządek realizowanych połączeń (od najdłuższych lub najkrótszych), rodzaj danych uwidocznionych na ekranie oraz pary warstw, dla których następuje prowadzenie ścieżek. Dla każdej pary, ścieżki prowadzone są (w zasadzie) poziomo na pierwszej warstwie i pionowo na drugiej. Możliwe jest zadanie różnych szerokości ścieżek dla maksymalnie 10 sieci o określonych nazwach. W trakcie automatycznego trasowania, niezrealizowane połączenia są nanoszone na dodatkową warstwę, która wraz z raportem końcowym stanowi materiał wyjściowy do uzupełnienia projektu przez operatora.
Fazę projektowania pakietów elektronicznych kończy program PC-PHOTO. Służy on do realizacji zbiorów sterujących fotoplotterami: GERBER, GTCO i FLASHSCAN. W programie zostaje określony zbiór numerów przesłon i ich wymiarów, a także rodzaj zapisu współrzędnych w zbiorach sterujących (przyrostowy lub absolutny) oraz współrzędne początku zapisu oraz podziałka wykonania kliszy. Fotoplotery GTCO i FLASHSCAN mogą być podłączone do komputera za pomocą łącza transmisji szeregowej. Niektóre fotoplotery firmy GERBER wymagają przekopiowania zbioru sterującego na taśmę magnetyczną.
Po wykonaniu projektu obwodu drukowanego, dokonane w nim zmiany, związane z rozmieszczeniem i wyborem elementów, są automatycznie nanoszone na schemat ideowy za pośrednictwem programu PC-BACK. Porównuje on pierwotny raport utworzony przez program PC-PACK, z raportem zamian dokonanych w trakcie pracy z programem PC-CARDS lub PC-HISTO i zakłada zbiór informacji zwrotnych. Zbiór ten jest zestawem poleceń realizowanych przez program PC-CAPS, w celu automatycznego zmodyfikowania schematów.
Trzecim blokiem funkcjonalnym systemu P-CAD jest program PC-LOGS. Jest to konwersacyjny program symulujący działanie układów cyfrowych i rozróżniający 12 stanów logicznych. Używany jest łącznie z programem PC-CAPS oraz z translatorami listy połączeń. Umożliwia on zweryfikowanie poprawności logicznej projektowanego układu przed wykorzystaniem w projekcie obwodu drukowanego listy połączeń.
Poszczególne bloki mogą być opisywane za pomocą bramek, jak i przez definicję funkcjonalną układu modelowego. Do funkcjonalnego opisu bloku układu służy język PML (opis języka dostarczany jest razem z systemem). Umożliwia on określenie zbioru warunków definiujących zależność sygnałów wyjściowych bloku (moduły) od sygnałów wejściowych. Jest to zbiór tekstowy. Po skompilowaniu przez program PC-MODEL, zbiór jest łączony z listą połączeń.
Projektant ma możliwość tworzenia wielowarstwowego lub hierchicznego projektu, realizowanego z bramek i typowych układów cyfrowych. Wykorzystywane są też specjalne symbole programu PC-LOGS, zawarte w bibliotece (modele strukturalne). Połączenie skompilowanego zbioru modelowego z listą połączeń odbywa się w trakcie działania programu PRESIM. W rezultacie otrzymywana jest zweryfikowana lista połączeń, stanowiąca podstawę do analizy logicznej projektu.
Projektant ma następujące możliwości: definiowanie wyjściowych sygnałów sterujących, obserwacja przebiegów w dowolnym punkcie układu, kontrola stanów niestabilnych w wybranym momencie. Możliwe jest też zbadanie pracy układu przy częstotliwościach granicznych.
Przegląd i analizę wyników uzyskanych w trakcie symulacji oraz określenie warunków zdarzeń umożliwia program POSTSIM. Warunek zdarzeń jest kombinacją stanów logicznych lub zmian w stanach logicznych jednego sygnału lub całej grupy. Dla określonego warunku POSTSIM identyfikuje stany całego układu.
Wyniki symulacji zapisywane są w postaci przebiegów lub tablicy prawdy i mogą być przedstawione na ekranie, zapisane do zbioru dyskowego lub wydrukowane. Tablicowa reprezentacja rezultatów symulacji podawana jest w kodzie binarnym, oktalnym lub szesnastkowym.
System P-CAD jest stale ulepszany. Niedawno zaprezentowano wzbogaconą wersję systemu. Wprowadzone zmiany zwiększają efektywność automatycznego prowadzenia ścieżek. Biblioteka została wzbogacona o elementy montażu powierzchniowego. Nowa wersja umożliwia też projektowanie jednostronnych obwodów drukowanych dla układów analogowych oraz wykorzystanie systemu w sieci komputerowej.

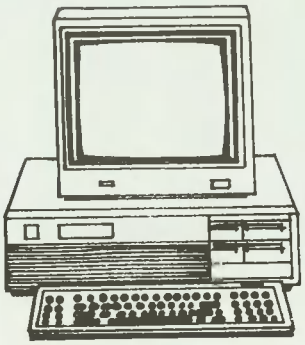
Ewa Marjańska





Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

INTERTMS



Sp. z o.o.
BIURO USŁUG
KOMPUTEROWYCH
00-867 WARSZAWA
ul. Chłodna 35/37
Tel. 24-78-23

IBM PC XT/AT, AMSTRAD

OFERUJEMY

✻WSZYSTKIE KONFIGURACJE SPRZĘTU
✻URZĄDZENIA PERYFERYJNE
— drukarki
— plotery
— digitizery
— streamery
— zabezpieczenia zasilania
— terminale
— myszy
— karty IBM PC XT/AT

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

OFERUJEMY

✻SYSTEMY POD KLUCZ
✻SIECI KOMPUTEROWE
✻SIECI WIELODOSTĘPU

OFERUJEMY

✻BOGATĄ BIBLIOTEKĄ OPROGRAMOWANIA
SPRAWDZONE PROGRAMY DLA:
— księgowości, płac i kadr
— kosztorysowania
— gospodarki materiałowej
— planowania inwestycji
— automatyzacji prac biurowych
— atestacji stanowisk pracy
— technicznego przygotowania produkcji (TPP)
— projektowania wspomaganego komputerem (CAD)
BĘDĄCE NASZYMI PRODUKTAMI

UDZIELAMY GWARANCJI
I PROWADZIMY SERWIS
W NASZYM SALONIE ORGANIZUJEMY
POKAZY I DEMONSTRACJE SPRZĘTU
I OPROGRAMOWANIA
KATALOGI Z NASZĄ OFERTĄ HANDLOWĄ
WYSYŁAMY POCZTĄ

INTERTMS

IBM jest znakiem firmowym International Business Machines Corp.

Więcej cali...

Stosowany przez użytkowników programu AutoCAD moduł obsługi drukarki typu Epson FX-80, PPEPS.DRV (Printer Plotter EPSon DRiVer), ogranicza wymiary rysunku do 8x11 cali. Tymczasem wiele drukarek kompatybilnych z drukarką Epson FX-80 ma szerszy wałek, zaś maksymalna wysokość rysunku zależy jedynie od rodzaju stosowanego papieru. Aby w pełni wykorzystać ich możliwości, warto więc zmodyfikować program obsługujący drukarkę.

Maksymalne wymiary drukowanego rysunku kodowane są w module PPEPS.DRV w dwojaki sposób. Wysokość jest podawana w calach i zapisana w postaci liczby zmiennoprzecinkowej typu LONG REAL (tab. 1), w standardowym formacie koprocesora INTEL 8087. W module PPEPS.DRV dostarczanym z wersją AutoCAD 2.18, liczba ta umieszczona jest poczynając od bajtu o adresie względnym 428H. Maksymalną szerokość określają dwa parametry:
- liczba punktów na cal w trybie graficznym dla danego typu drukarki (liczba całkowita zapisana w dwu bajtach, od adresu względnego 3BFH), typowo - 120 (78H);
- maksymalna liczba punktów w linii w trybie graficznym (liczba całkowita zapisana w dwu bajtach, od adresu względnego 3BDH).

Maksymalna szerokość rysunku wynika z podzielenia maksymalnej liczby punktów w linii przez liczbę (gęstość) punktów na cal (tab. 2).

Kody można zmienić za pomocą jednego z wielu dostępnych programów narzędziowych służących do edycji zawartości zbiorów (np. Norton Utilities, Debug, PC Tools itp.). Zmiany pozwalające uzyskać rozmiary drukowanego rysunku na 15.6 x 24 ilustruje wydruk.

Moduły obsługi innych drukarek (PP*.DRV) dostarczane wraz z programem AutoCAD, w większości wypadków również stosują opisaną metodę kodowania maksymalnych rozmiarów drukowanego rysunku.

Marek Bojda
Krzysztof Kotara

Tabela 1. Przykłady kodów wysokości rysunku

Wysokość w calach	Postać LONG REAL, kolejne bajty (szesnastkowo)							
8	00	00	00	00	00	00	20	40
10	00	00	00	00	00	00	24	40
12	00	00	00	00	00	00	28	40
15	00	00	00	00	00	00	2E	40
20	00	00	00	00	00	00	34	40

Tabela 2. Przykłady obliczania szerokości rysunku

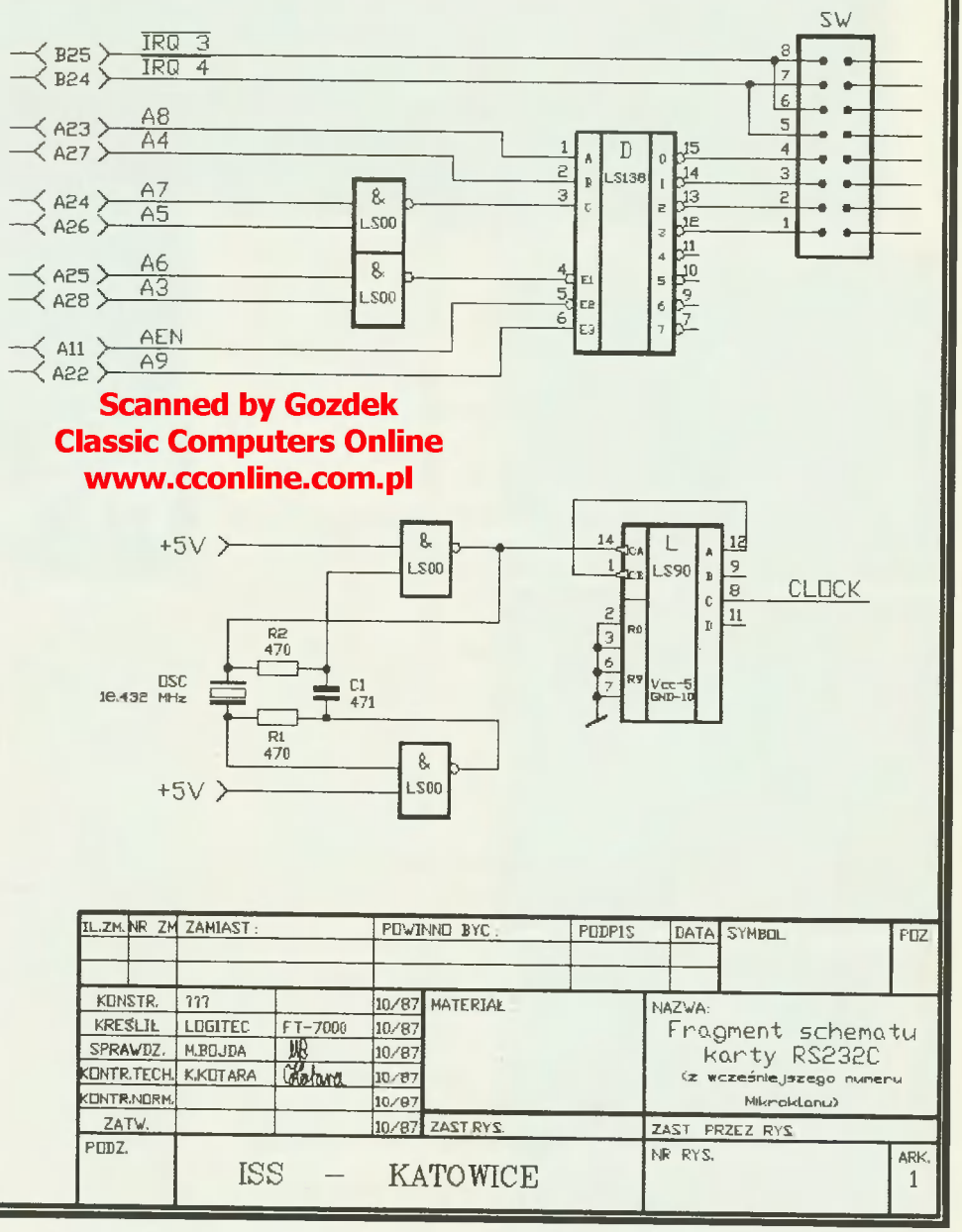
Drukarka	Liczba punktów w linii	Liczba punktów na cal	Szerokość w calach
Logitec FT-7000	1872	120	15,6
STAR SD-15	1632	120	13,6

Wydruk. Przykład modyfikacji PPEPS.DRV za pomocą programu DEBUG

Zapis operacji	Komentarz
debug ppeps.drv	ppeps.drv na domyślnym dysku
-s 100 5c1 c0 03	szukanie liczby punktów w linii
xxxx:04BD	adres podany przez DEBUG
-e 4bd 50 07	zamiana na 1872 punkty w linii
-s 100 5c1 00 00 00 26 40	szukanie wysokości w calach
xxxx:0528	adres podany przez DEBUG
-e 52e 38	zamiana na 24 cale
-w	zapis na dysk
Writing 04C2 bytes	
-q	zakończenie pracy

Elektronicy próbują wykorzystać AutoCAD do tworzenia schematów ideowych. Aby zadanie to ułatwić, tworzone są biblioteki zawierające standardowe symbole graficzne typowych układów. Biblioteka taka została utworzona też i w Polsce, w Instytucie Systemów Sterowania. Przyjęte w niej symbole odpowiadają normie branżowej, przygotowanej z myślą o automatyzacji kreślenia schematów. Zastosowane w niej symbole „ułatwiają życie" urządzeniom wykonawczym, ale są niezbyt przejrzyste. Sposób oznaczania układów cyfrowych (funktorów logicznych) stosowany w USA jest o wiele wygodniejszy dla użytkownika - łatwiej można przeanalizować działanie układu. Nowoczesne drukarki bez specjalnych trudności rysują zaokrąglenia w trybie graficznym. W ploterach, łuki są tworzone z krótkich odcinków, co również daje niezłe rezultaty. Dlatego też na Zachodzie powstały biblioteki symboli układów elektronicznych według norm amerykańskich.
Program AutoCAD został pomyślany jako uniwersalne narzędzie dla projektanta - architekta, mechanika itp. W stosowaniu go do tworzenia schematów ideowych, przypomina jednak przysłowiowe wytaczanie armaty na wróble. Znacznie lepszy do tego celu wydaje się być program OrCAD, którego dodatkową zaletą jest możliwość przeniesienia wprowadzonych danych do systemów projektowania płytek drukowanych. W jednym z najbliższych numerów Mikroklanu opublikujemy więcej informacji o programie OrCAD.

(red).



Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

IL.ZM.	NR ZM	ZAMIAST:		POWINNO BYC:	PODPIS	DATA	SYMBOL	POZ

KONSTR.	???		10/87	MATERIAŁ	NAZWA: Fragment schematu karty RS232C (z wcześniejszego numeru Mikroklanu)
KREŚLIŁ	LOGITEC	FT-7000	10/87		
SPRAWDZ.	M.BOJDA		10/87		
KONTR.TECH.	K.KOTARA		10/87		
KONTR.NORM.			10/87		
ZATW.			10/87	ZAST.RYS.	ZAST. PRZEZ RYS.
PODZ.	ISS – KATOWICE				NR RYS. ARK. 1

mikrokram

Business is Business

Wielu producentów sprzętu komputerowego nie zajmuje się ... jego sprzedażą. Wygodniej jest - szczególnie na obcym rynku - znaleźć firmę z odpowiednim doświadczeniem handlowym czyli tzw. dystrybutora. Dystrybutor organizuje odpowiednią promocję, service i sieć handlową. Do renomowanych dystrybutorów sprzętu komputerowego na rynku RFN zaliczana jest firma Synelec. Z jej usług korzysta m. in. Seagate (dyski sztywne), Pertec (pamięci masowe), Archive (pamięci taśmowe), Maxell (dyskietki), Esprit (terminale) i Citizen (drukarki). Do nowości technicznych. które oferuje Synelec należą jednokrotnie zapisywalne pamięci optyczne ISI o pojemności 115/230 MB współpracujące z komputerami PC XT/AT. Zainteresowanie, z jakim spotkały się drukarki Citizen na rynku zachodnioeuropejskim, skłoniło firmę Synelec do poszukiwania dalszych rynków zbytu. Niedawno uzyskała ona uprawnienia do pełnego reprezentowania interesów i dystrybucji produktów Citizena na rynkach wszystkich krajów RWPG. W związku z tym prowadzone są rozmowy na temat montażu drukarek Citizena na terenie Polski. Synelec zamierza m. in. wprowadzić na polski rynek drukarkę laserową Citizena ... z polskimi znakami diakrytycznymi. Ewentualny import podzespołów do Polski realizowany byłby na zasadach handlu kompensacyjnego. inf. własna **(AJP)**

Windows/386

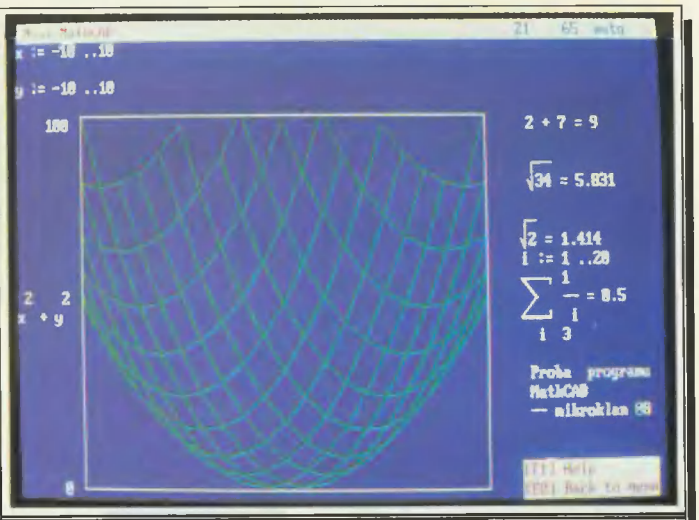
Firma Microsoft wprowadziła nową wersję popularnego programu rozszerzającego funkcje systemu operacyjnego MS-Windows - przeznaczoną dla systemów 32-bitowych. Windows/386 zapewnia możliwość współbieżnego przetwarzania programów i wykorzystuje mechanizmy systemu wirtualnego procesora 80386. Możliwe jest więc stosowanie oprogramowania stworzonego do współpracy z wcześniejszymi wersjami Windows w komputerach PC 386 i PS/2 Model 80, przy pełnym wykorzystaniu możliwości procesora nowej generacji. Programy przeznaczone do pracy pod kontrolą MS-DOS mają zapewnioną możliwość korzystania z 640 KB wirtualnej pamięci operacyjnej. Dla programów. które zapisują informacje bezpośrednio do pamięci obrazu przewidziano mechanizm nakładających się okien na ekranie. System, w którym stosowany jest program Windows/386 musi zawierać co najmniej 2 MB pamięci operacyjnej.
Nowa wersja systemu kosztuje 150 funtów angielskich. Dotychczasowi użytkownicy programu mogą zakupić nową wersję za 75 funtów. Byte 12/87 **(kwp)**

VGA także dla AT

Nowa karta PC VGA firmy Paradise umożliwia uzyskanie grafiki zgodnej z nowym rozwiązaniem VGA (zastosowanym przez IBM dla PS/2) także w komputerach typu PC AT. Karta realizuje tryb graficzny VGA z rozdzielczością 800x600 punktów przy 16 kolorach wybieranych z palety 256 możliwych. W trybie alfanumerycznym można uzyskać 132 kolumny. PC VGA została skonstruowana m. in. w oparciu o układ PVGA firmy Paradise i układ 80-MHz-Pallete firmy Brooktree. Posiada 512 KB pamięci video RAM i 32 KB pamięci ROM. Zapewnia maksymalną rozdzielczość 1024x768 punktów z 4 kolorami. Jest w pełni kompatybilna z kartami typu EGA, CGA, MDA i Hercules.
Karta firmy Paradise nie wymaga stosowania specjalnych procedur sterujących - całe oprogramowanie pracujące na standardowych kartach firmy IBM i karcie Hercules powinno działać z nową kartą bez zastrzeżeń. Paradise dostarcza jednak procedury pozwalające w pełni wykorzystać możliwości karty przez programy, takie jak np. Windows, GEM, Page Maker, Ventura Publisher, Lotus 1-2-3, Framework II, Microsoft Word i inne. Cena PC VGA - 599 USD. Byte 11/87 **(as)**

Żywa tablica

Wykorzystanie komputera jako pomocnika w rozwiązywaniu zagadnień wymagających obliczeń zmuszało naukowców do zmiany przyzwyczajeń i sposobu pracy. Bezduszna maszyna żądała programów i algorytmów... Okazuje się, że może być inaczej.



Program MathCAD firmy MathSoft jest połączeniem elektronicznego kalkulatora, programu CAD, procesora tekstów i wyspecjalizowanego języka programowania. Ekran monitora przekształcono w inteligentną tablicę do rozwiązywania zadań. Użytkownik może pisać wzory i deklarować zmienne w dowolnym miejscu pola roboczego - ekran stanowi okno, w którym widać aktualnie używaną część tego pola. Rozmiary pola roboczego są ograniczone tylko wielkością dostępnej pamięci - autorzy programu twierdzą, że wymagane minimum - 384 KB - wystarcza w większości sytuacji.

Stosowane w programie operatory matematyczne są takie same, jak w klasycznej matematyce - np. znakiem mnożenia jest kropka, dzielenie zapisywane jest jako kreska ułamkowa z czynnikami pod i nad nią. Kolejność zapisywania działań jest również zgodna ze zwyczajami. Można używać sumowania (znak sigma), greckich liter i wieloznakowych nazw zmiennych.

Program pozwala też w klasyczny sposób zapisywać zmienne indeksowane. Wyniki prostych działań pojawiają się na ekranie niemal natychmiast po napisaniu znaku równości. Zakres liczb, jakimi MathCAD może operować, jest imponujący - np. największa liczba całkowita jest równa 10 do potęgi 308, a format jej zapisu na ekranie jest definiowany przez użytkownika. Można posługiwać się notacją naukową (mantysa i wykładnik) lub zwykłą (określając liczbę cyfr przed i po kropce). Wyniki działań mogą być podawane w postaci wykresów, umieszczanych we wskazanym miejscu pola roboczego. Wykresy można dowolnie skalować, dobierać zakresy zmiennych itp.

MathCAD umożliwia pełnoekranową edycję tekstów i wyrażeń matematycznych: dostępne są wszystkie typowe funkcje procesora tekstów. Pewnym utrudnieniem dla użytkownika może być jednak nietypowe zachowanie się kursora - pozostaje on na pozycji ostatnio wpisanego znaku (zamiast wskazywać następną pozycję). Konieczne jest też zapamiętanie znaczenia poszczególnych klawiszy przy posługiwaniu się symbolami matematycznymi i literami greckimi. Teksty wypisywane na ekranie mogą być elementami rozwiązania lub komentarzami.

Komentarze, pomijane przez program przy obliczeniach, wpisuje się rozpoczynając i kończąc tekst cudzysłowami (cudzysłowy nie są wyświetlane na ekranie).

Elastyczność w operowaniu obrazem autorzy MathCAD-u uzyskali posługując się trybem graficznym o dużej rozdzielczości (monochromatycznym). Program może współpracować z kartami graficznymi CGA i Hercules. Nieco ograniczone są, niestety, możliwości korzystania z drukarki - można używać tylko drukarek graficznych IBM i HP LaserJet.

MathCAD dopuszcza stosowanie trzech typów zmiennych:
- zmienne skalarne - mające pojedynczą wartość
- zmienne indeksowane, w których każdemu indeksowi jest przypisywana pewna wartość (podobnie jak w jednowymiarowej tablicy)
- zmienne zakresowe - obejmujące wartości z pewnego zakresu (np. wszystkie liczby całkowite i ułamkowe z zakresu 1... 10)

W programie występują dwa unikalne operatory: **until** i **size**. Mogą być stosowane - łącznie ze zmiennymi zakresowymi i indeksowanymi - do ustalenia liczby powtórzeń w pętlach obliczeniowych.

Pakiet ma wbudowane wszystkie podstawowe funkcje matematyczne: trygonometryczne, hiperboliczne i logarytmiczne, Bessela i funkcje zespolone pozwalające m. in. wyznaczyć rzeczywiste i urojone części liczb zespolonych. Istnieje też możliwość definiowania funkcji przez użytkownika.

Zmienne mogą być lokalne lub globalne. Ponieważ program odczytuje pole robocze zgodnie z klasyczną konwencją (od góry do dołu i od lewej do prawej), dlatego obszar, w którym określana jest zmienna lokalna zależy od miejsca jej zdefiniowania. Musi być ona zdefiniowana przed wyrażeniem, w którym jest używana. Zmienne globalne mogą być definiowane w dowolnym miejscu.

Dużą zaletą MathCAD-u jest to, że rozpoznaje on jednostki miar(stopnie, metry, kilogramy itp.). W czasie obliczeń sprawdzana jest zgodność jednostek i ustalana jednostka wyniku (np. jeśli pomnożona zostanie szybkość przez czas - MathCAD poda wynik w jednostkach miary odległości; dodając np. cale do metrów nie trzeba troszczyć się o właściwe przeliczniki). Program nie ma wprawdzie żadnych wbudowanych jednostek, lecz można z łatwością zdefiniować jednostki i zależności między nimi, używając zmiennych globalnych. Wraz z programem dostarczane są też zbiory zawierające definicje kilku systemów jednostek.

MathCAD ma dwa tryby pracy: umożliwiający bezpośrednie wprowadzanie danych oraz rozkazowy. Program przechodzi do trybu rozkazowego po naciśnięciu klawisza **Esc**, co powoduje wyświetlenie zgłoszenia (ang. prompt) na górze ekranu. Niektóre rozkazy są przypisane klawiszom funkcyjnym, dzięki czemu można wywoływać je w dowolnej chwili naciśnięciem pojedynczego klawisza. Do operacji na zbiorach przewidziano trzy komendy: **Save**, **Load** i **Append**. MathCAD umożliwia też stosowanie poleceń DOS-u bez konieczności opuszczenia programu.

Wykorzystanie okien pozwala używać dwóch zbiorów jednocześnie. Odpowiedni rozkaz powoduje pionowy podział ekranu na dwie części, z których jedna jest zamrażana, a druga aktywna (dotyczą jej wprowadzane rozkazy). Za pomocą odpowiedniej kombinacji dwóch klawiszy można zmieniać wzajemne proporcje dwóch okien lub przełączać okna aktywne.

Podobnie jak w większości arkuszy obliczeniowych, MathCAD zapewnia pracę w sterowanym lub automatycznym trybie obliczeń. W trybie automatycznym wszystkie obliczenia, wykresy itd. są wykonywane na bieżąco (jeśli zostanie wpisany nowy wzór lub obszar roboczy zostanie przesunięty tak, że ukaże się poprzednio niewidoczne równanie - program automatycznie obliczy nowe wartości). Niestety, nawet pracując na komputerze PC AT z zegarem 12 MHz i koprocesorem arytmetycznym trzeba niekiedy czekać na zakończenie tych operacji. Gdy rozwiązuje się trudny problem, wymagający uzyskania wielu wyników pośrednich, wygodniej jest używać trybu sterowanego, w którym obliczenia wykonywane są dopiero po wydaniu właściwego rozkazu.

Nie należy oczekiwać, że MathCAD zastąpi podręczne kalkulatory ekranowe (jak np. SideKick), lecz nowa koncepcja posługiwania się komputerem zaprezentowana w tym programie z pewnością zasługuje na rozpowszechnienie. Połączenie znacznej mocy obliczeniowej, obsługi zgodnej z intuicją człowieka i korzystanie z zasady WYSIWYG (ang. What You See Is What You Get) sprawia, że MathCAD jest komfortowym i wydajnym narzędziem pracy. Nie zmusza on użytkownika do pozbywania się starych nawyków i uczenia się nowego stylu pracy z komputerem. Można przypuszczać, że wkrótce pojawią się inne programy użytkowe oparte na podobnych zasadach - być może MathCAD stanie się protoplastą nowej jakości w dziedzinie problemowo zorientowanych programów.

Opr. Janusz Wrześniak wg PC Week 11/86

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

AutoCAD 2.6

Ogólny opis systemu AutoCAD był już publikowany w **mikro**klanie (nr 1/87). Dlatego też niniejszy artykuł stanowi jedynie uzupełnienie podanych informacji o cechy wprowadzone do wersji 2.6 (siódmej z kolei od chwili ukazania się pakietu na rynku). Przedstawiamy też program AutoSKETCH — uproszczoną wersję AutoCad-u

Na wystawie CADCAM '87, w Birmingham, firma Autodesk Inc. zaprezentowała nową wersję pakietu do komputerowego wspomagania prac projektowych AutoCAD. Od wprowadzenia w 1982 roku pierwszej wersji, produkt ten zyskuje coraz więcej zwolenników. Świadczy o tym liczba 80 tys. instalacji, stanowiących ok. 44 % ogólnej liczby sprzedanych systemów typu CAD. Nie tylko wskaźniki liczbowe, lecz także wysokie oceny funkcjonalności i efektywności pakietu, świadczą o tym, że stał się on już standardem w swej klasie.

Jednym z najbardziej znaczących rozszerzeń nowej wersji pakietu jest wprowadzenie grafiki trójwymiarowej (3D). Daje to niewątpliwie pożyteczną możliwość obrazowania projektowanego obiektu w sposób bardziej realistyczny, niż miało to miejsce w stosowanym dotąd płaskim planie.

Linie, punkty i powierzchnie mogą być definiowane w trójwymiarowej przestrzeni w oparciu o współrzędne podawane przez operatora z klawiatury lub digitizera, wprowadzane ze zbiorów w formacie **.DXF** lub przetworzone w procedurach AutoLISP-u.

Trójwymiarowe obiekty mogą być tworzone przy wykorzystaniu podstawowych figur geometrycznych , jak kule, sześciany, stożki, walce czy spirale. Całość konstruowanego obiektu może być rozciągana, powiększana lub zmniejszana. Można również dokonywać zmian w poszczególnych elementach rysunku, przy czym wszystkie wymiary związane z modyfikowanym wymiarem są korygowane automatycznie.

Jest to możliwe dzięki wprowadzeniu zasady wymiarowania (ang. associative dimensioning), która umożliwia jednoczesną kontrolę wszystkich związanych ze sobą parametrów, przez co nie ulegają zachwianiu właściwe proporcje rysunku. I tak polecenie **EXPLODE** zostało rozszerzone o możliwość rozdzielenia wymiarów całości kreślonego obiektu na poszczególne części składowe, jak linie, strzałki i tekst.

UPDATE pozwala zmienić całość wymiarów przy wykorzystaniu aktualnych zmiennych wymiarowych, stylu opisu czy jednostek.

HOMETEXT przywraca początkową pozycję i rozmiar tekstu opisu, zmienione wcześniej np. przy pomocy funkcji **STRETCH** (rozciągnięcie).

NEWTEXT pozwala zastąpić wszystkie opisy wymiarowe nowym tekstem.

Funkcje AutoCAD-u, jak **ZOOM** (powiększanie i zmniejszanie), **PAN** (przesuwanie części rysunku bez zmiany skali) i **VIEW** (podgląd), działają w nowej wersji w sposób przezroczysty, tzn. niezależny od komend rysowania i edycji rysunku. Można więc rozpocząć kreślenie w jednym miejscu, potem zastosować polecenia **PAN** lub **ZOOM** i wrócić do poprzedniej czynności.

Funkcja **AREA** została rozszerzona o możliwość obliczania powierzchni koła czy wieloboku przez zwykłe wskazanie kursorem na obiekt (tak jak w instrukcji **MOVE**). AutoCAD 2.6 pozwala również obliczać sumę powierzchni kół i wieloboków, a także dodawać i odejmować poszczególne powierzchnie.

Obraz utworzony przy pomocy pakietu może być oglądany z dowolnej perspektywy, przy czym linie zasłonięte usuwa się przy pomocy funkcji **HIDE**. Dużą zaletę stanowi też fakt, że płaskie rysunki, opracowane przy użyciu poprzednich wersji AutoCAD-u (z którymi nowa wersja jest w pełni kompatybilna), mogą być przenoszone w przestrzeń trójwymiarową.

Tak jak poprzednie wersje, również obecna wykorzystuje wbudowany interpreter języka Lisp (AutoLISP), który umożliwia dostęp do danych w tablicy symboli, takich jak: warstwy, typy linii czy bloki. Program ten został jednak znacznie wzbogacony, między innymi o możliwość wykorzystania zalet grafiki trójwymiarowej.

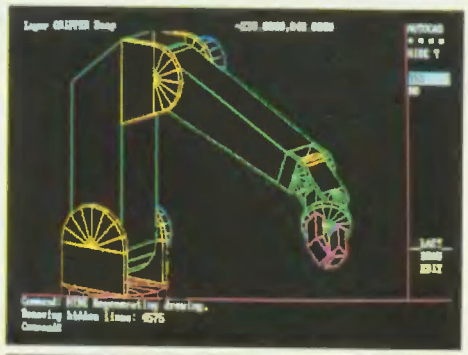
Zestaw nowych, rozszerzonych instrukcji programu uzupełniają polecenia służące do współpracy z innymi pakietami wspomagającymi, opracowanymi przez firmę Autodesk (np. funkcja **FILMROLL**, pozwalająca utworzyć zbiory wykorzystywane przez pakiet AutoSHADE) oraz funkcje (i zmienne) związane ze środowiskiem (np. zmienne zarządzające częścią rozszerzonej pamięci komputera typu PC AT, co pozwala na bezkolizyjne współistnienie AutoCAD-u z innymi programami wykorzystującymi rozszerzenia pamięci).

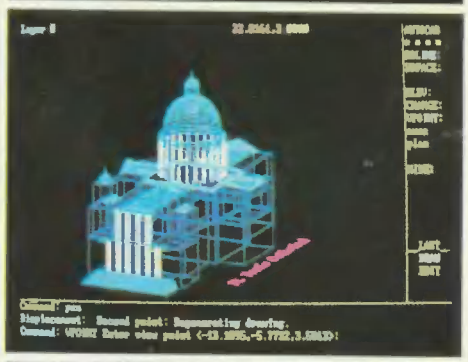
Według zapewnień producenta, AutoCAD jest w pełni przystosowany do komputerów z nowej serii IBM PS/2 (dostarczany jest między innymi na dyskach 3,5" o pojemności 720 KB). Może współpracować z systemem graficznym VGA (Video Graphics Array), który zapewnia rozdzielczość 320x200 punktów w 256 kolorach lub 640x480 punktów w 16 kolorach, a przy zastosowaniu pakietu 8541A pozwala uzyskać rozdzielczość 1024x768 punktów w 256 kolorach.

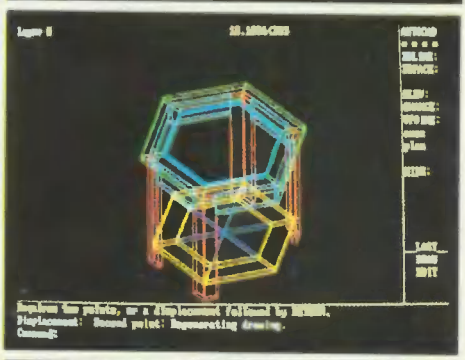
AutoCAD dla architektów

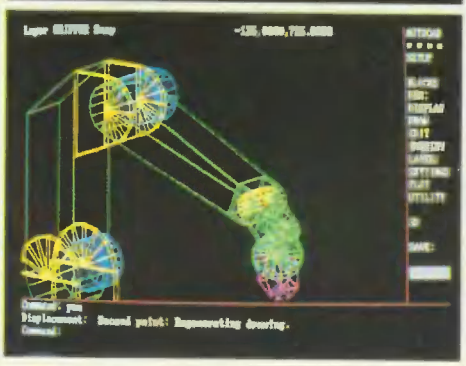
AutoCAD AEC Architectural to specjalizowany pakiet programów adresowany do użytkowników systemów CAD branży budowlanej: od projektantów do dekoratorów wnętrz.

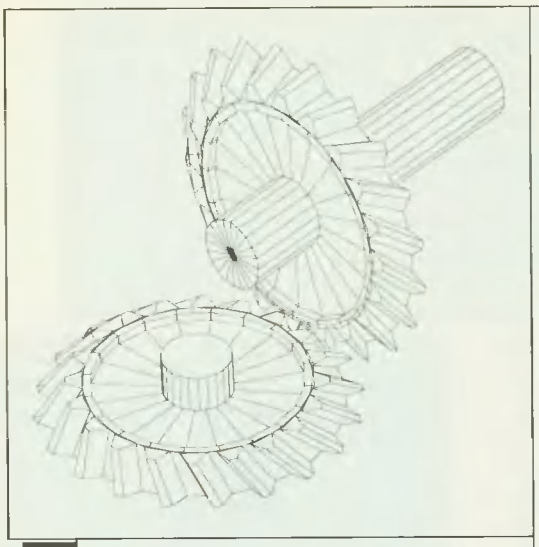
Powstał on w oparciu o dezyderaty praktyków z tej dziedziny, dzięki czemu został dobrze przystosowany do jej specyficznych zadań i wymogów.



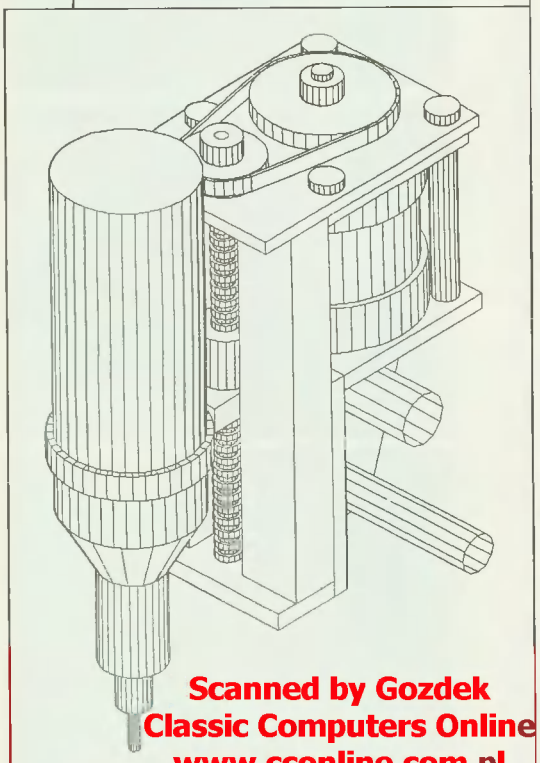








Zautomatyzowanie wielu rutynowych czynności jak też możliwość wykorzystania w opracowanym projekcie gotowych już elementów rysunku, pozwala przy pomocy programu AutoCAD AEC zaoszczędzić wiele godzin żmudnej pracy.
Pakiet programów zawiera między innymi bibliotekę ponad 250 symboli stosowanych w kreślarstwie takich, jak detale konstrukcyjne, elementy wykończeniowe, akcesoria i wyposażenie, osprzęt elektryczny i mechaniczny, urządzenia sanitarne itp. Wszystkie symbole graficzne mogą być dostosowane



Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

do indywidualnych wymogów i w miarę potrzeby uzupełniane o inne, opracowane przez użytkownika i włączone do biblioteki symboli.
Możliwe jest także przypisanie im określonego miejsca w tabelach symboli i rozkazów (stanowiących rodzaj Menu), by ułatwić ich stosowanie w trakcie pracy.
Zastosowanie makroinstrukcji, grupujących wiele komend w funkcjonalne zespoły (np. wprowadzane symbole są otwierane w planie zerowym a następnie automatycznie skalowane i umieszczane we właściwym miejscu jedną instrukcją) znacznie upraszcza projektowanie, a jednocześnie zapobiega niedopasowaniu wielkości tekstu czy symboli do przeznaczonej dla nich przestrzeni.
Wykorzystanie procedur języka Lisp jest jedną z największych zalet pakietu AutoCAD AEC Architectural, która usprawnia i automatyzuje wiele czasochłonnych czynności kreślarskich. Przy jego pomocy np. podwójne linie murów i wszelkie zagłębienia trasuje się używając pojedynczej kreski. Operacje wkomponowywania drzwi i okien w wykonanym już zarysie muru są łatwe i szybkie. Rodzaj drzwi jest wybierany przy pomocy menu a miejsce wstawienia wskazuje się kursorem. Program automatycznie robi wyłom w murze i wstawia wybrany element. Jeśli pozycja drzwi jest później zmieniana, nieciągłości w murze są automatycznie korygowane.
Również w sposób zautomatyzowany są kreślone żebrowania stropów (obliczone uprzednio dla poszczególnych pomieszczeń nawet na skomplikowanych rysunkach).
Pracując w trybie konwersacyjnym, procedury Lispu zgłaszają użytkownikowi do wyboru różne opcje, jak np. rozmiary, kąt nachylenia czy specjalny rodzaj konturów.

AutoSKETCH dla debiutantów i zaawansowanych

Jest to tani (ok. 80 funtów) pakiet oferujący wiele standardowych funkcji kreślarskich, jak trasowanie linii, prostokątów, kół, łuków itp. Daje także możliwość wprowadzania tekstu, który jak całość rysunku może być transformowany przy pomocy prostych poleceń takich, jak **ZOOM**, **PAN**, **STRETCH**, **MIRROR** (zwierciadlane odbicie), **SCALE** (skalowanie) czy **ROTATE** (obrót). Potrafi mierzyć linie, powierzchnie i kąty (maksymalna dokładność do 7 miejsc po przecinku).
W programie może być także wykorzystywana zasada associative dimensioning, co znacznie upraszcza korektę lub modyfikację danych. Obiekty mogą być rozmieszczone na dziesięciu warstwach, z których każda może być niezależnie opracowywana i poprawiana, zapisywana na dysku i stamtąd wykorzystywana w innych rysunkach. Pakiet zawiera również bogatą bibliotekę symboli.
AutoSKETCH może być wykorzystywany na komputerach zgodnych z PC XT lub AT, z kartą EGA i kolorowym monitorem. Przy monitorze monochromatycznym, może współpracować z kartą CGA oraz Hercules. Możliwe jest korzystanie z koprocesora arytmetycznego oraz wyjście na drukarki mozaikowe lub laserowe, jak też plotter. Istnieje także możliwość tworzenia zbiorów w formacie **.DXF** (do wykorzystania w programie AutoCAD) oraz slajdów w formacie **.SLD**, stosowanych w systemach DTP.
Wiele cech tego programu, jak np. łatwość opanowania zasad obsługi, czyni z niego idealny wstęp do AutoCAD-u. Z drugiej zaś strony, jego bogate możliwości graficzne i niemała precyzja przy relatywnie niskiej cenie, czynią z niego poważną alternatywę dla tych użytkowników, dla których nabycie pakietu AutoCAD w pełnej konfiguracji (pełna cena wg PC Magazine z marca 1986 r. - 2500 $) byłoby zakupem na wyrost.

AutoSHADE - coś dla estety

Jedną z najnowszych wśród ponad trzystu propozycji zawartych w katalogu zastosowań, jest program pozwalający przetworzyć graficzne projekty (szkice, rysunki) w artystyczną interpretację pierwowzoru. AutoSHADE pozwala stworzyć realistyczny, kolorowy obraz obiektu, zachowujący głębię ostrości i perspektywę.
Proces cieniowania jest dwuetapowy. Najpierw tworzony jest przy pomocy AutoCAD-u 2.6 trójwymiarowy rysunek projektu. Następnie określa się pewną liczbę źródeł światła, tworzących rozproszone oświetlenie całego obiektu lub też punktowe oświetlenie z jednego kierunku.
Wreszcie ustala się punkt obserwacji w trójwymiarowej przestrzeni (używając polecenia **LOOKFROM**) oraz punkt obserwowany, czyli punkt, na który jest skierowany obiektyw (polecenie **LOOKAT**).
Można w ten sposób przygotować jedno lub kilka ujęć obserwowanego przedmiotu i następnie zarejestrować je na dysku przy pomocy opcji **FILMROLL**. Zapamiętany w ten sposób obraz może być wyświetlony na monitorze kolorowym i utrwalony na zdjęciu lub slajdzie, a także wydrukowany na kolorowym urządzeniu graficznym.
W AutoSHADE efekt może być korygowany poprzez zmianę intensywności światła omiatającego przedmiot, zmianę ustawienia „soczewki obiektywu" oraz ogólnie przez zmianę wyglądu zewnętrznego obiektu. Jedynie pozycja światła nie może być zmodyfikowana przez ten program. Parametr ten musi być ponownie określony przy użyciu programu podstawowego (AutoCAD 2.6).
Użytkownicy AutoSHADE-u mogą więc tworzyć w pełni kolorowe, cieniowane wizje swoich projektów, pozwalające ocenić estetyczne walory proponowanych rozwiązań. Taki sposób prawie natychmiastowej, realistycznej wizualizacji nakreślonego obiektu, z pewnością znajdzie wielu zwolenników wśród pracowników marketingu i promocji w biurach projektów (cena z trzeciego kwartału 1987 - 500 funtów).

AutoCAD i astrokartografia

Jednym z najnowszych zastosowań systemu jest opracowywanie map gwiezdnych. Kanadyjczyk Alan W. Paeth z Uniwersytetu w Waterloo stworzył program wykreślający mapy nieba na podstawie „Yale Bright Star Catalogue", uwzględniające ok. 9000 gwiazd do 8 rzędu wielkości (najciemniejsze gwiazdy widoczne gołym okiem są 6-7 wielkości). Natomiast pracownik firmy Autodesk napisał inny program, który przetwarza mapy gwiezdne Paeth'a w zbiory w formacie AutoCAD (**.DXF** files).

opr. **Antoni Szendzielorz**
wg materiałów firmowych AutoDesk.

Analyser

Dzięki ułatwieniom, jakie niesie ze sobą zwiększenie możliwości mikrokomputerów, specjalistyczne programy typu CAD stają się dostępne dla użytkowników popularnego sprzętu. Przykładem może być pakiet programów do analizy liniowych układów elektronicznych brytyjskiej firmy Number One Systems Ltd. Warto jednak zwrócić uwagę, że choć Analyser może ułatwić realizację wielu projektów, te wersje przeznaczone dla komputerów domowych trudno traktować jako narzędzie pracy dla profesjonalistów (red.)

Analyser I i Analyser II stosujące analizę węzłową, opartą na metodzie Kirchhoffa, umożliwiają określenie parametrów układów biernych i czynnych: wzmocnienia (lub tłumienia), przesunięcia fazowego, impedancji wejściowej i wyjściowej oraz opóźnienia grupowego. Ponadto Analyser II umożliwia analizę kryteriów stabilności oraz badanie układów generacyjnych (tab. 1). Stopień skomplikowania badanych układów jest ograniczony użytkowanym typem mikrokomputera. Kryterium określającym złożoność układu jest liczba węzłów (punktów połączeń poszczególnych elementów) oraz liczba i rodzaj elementów układu (tab. 2).

Obliczenie wzmocnienia oraz przesunięcia fazowego dla przedstawionego na rysunku przykładowego układu korektora wymaga (przy użyciu programu Analyser I i IBM PC XT) ok. 2 sekund dla każdego analizowanego punktu charakterystyki. Obliczenie opóźnienia grupowego dwukrotnie przedłuża czas analizy obwodu.

Menu programu (niezależnie od wersji) umożliwia:
- utworzenie nowego układu połączeń,
- modyfikację istniejącego układu,
- analizę parametrów układu znajdującego się w pamięci.
- zmianę analizowanego parametru (zastąpienie charakterystyki wzmocnienia w funkcji częstotliwości przez charakterystykę impedancji wejściowej lub wyjściowej),
- zapis na dyskietce i odczyt utworzonego układu połączeń,
- zmianę nazwy układu znajdującego się w pamięci,
- utworzenie na dyskietce katalogu układów.

Tworzenie układu połączeń polega na przyporządkowaniu kolejnych numerów węzłom układu (linie zasilania powinny mieć ten sam numer węzła) oraz na wprowadzeniu do pamięci komputera wartości elementów znajdujących się między węzłami. Wprowadzenie rezystora R1 o wartości 4700 Ω, znajdującego się między węzłami 1 i 2 przebiega w programie Analyser I następująco:

R1 RETURN 1 RETURN 2 RETURN 4.7E3 RETURN

Program Analyser II daje użytkownikowi więcej swobody - ten sam rezystor może być definiowany w jeden z następujących sposobów:

R1,1,2,4k7 RETURN
R1 1 2 4700 RETURN
R1 RETURN 1 RETURN 2 RETURN 4.7E3 RETURN
R1 1,2 .0047M RETURN
R1 1,2,M0047 RETURN

Po wprowadzeniu wartości elementów, należy jeszcze wcisnąć klawisz P i podać numery węzłów, będących kolejno wejściem, wyjściem i punktem wspólnym (masą) układu. Program wyświetla teraz listę elementów, a wciśnięcie dowolnego klawisza powoduje powrót do menu.

Utworzony lub wczytany z dyskietki układ może być modyfikowany lub poprawiony (pozycja 2 menu). Modyfikacja polega na ponownym zdefiniowaniu elementów znajdujących się między węzłami lub ponownym wprowadzeniu ich wartości.

W ten sam sposób mogą być zmieniane węzły stanowiące wejście-wyjście lub punkty przyłączenia masy.

Główną częścią programu jest analiza parametrów układu (pozycja 3 menu). Program wyświetla pytanie o liczbę analizowanych punktów oraz o ich częstotliwość początkową i końcową. Punkty charakterystyki można rozmieścić w analizowanym paśmie liniowo lub logarytmicznie. W paśmie częstotliwości akustycznych (od 20 Hz do 100 kHz), można również wybrać rezydentny zestaw częstotliwości (w sekwencji 1,2,5,7...).

Charakterystyki częstotliwościowe badanego układu mogą zawierać wzmocnienia bezwzględne lub względne, podane w skali liniowej lub logarytmicznej. Możliwe jest również otrzymanie składowej rzeczywistej i urojonej wzmocnienia (program dla ZX Spectrum nie ma tej opcji).

W podobny sposób może być wyliczana charakterystyka impedancji wejściowej lub wyjściowej w funkcji częstotliwości. Wymaga to przejścia do punktu 4 menu i zastąpienia słowa kluczowego **GAIN** przez odpowiednio **Zin** lub **Zout**.

Obie wersje programu Analyser bazują na uproszczonych modelach obliczeniowych elementów układu, przy czym Analyser II uwzględnia dodatkowo parametry rozproszone.

Zestawienie parametrów elementów, uwzględnianych przez każdą z wersji programu, przedstawiono w tabeli 3. Kiedy parametry elementu (np. rezystancja wyjściowa lub pojemność różnicowa wejść wzmacniacza operacyjnego) przyjęte przez program różnią się od zakładanych przez konstruktora, można je skorygować przez wprowadzenie do układu dodatkowych elementów zewnętrznych. Aby to ułatwić, program Analyser II umożliwia założenie własnej biblioteki elementów aktywnych. Korzystając z tej możliwości, każdemu typowi elementu należy przyporządkować właściwe oznaczenie, np. tranzystor **QA** może oznaczać **BF 199**, **QB 2N 3055**, wzmacniacz operacyjny **AA** oznacza **TL 080**, **AB - LM 324** itp. Umieszczenie na liście elementów układu oznaczenia typu elementu (zawartego w bibliotece) spowoduje automatycznie pobranie przez program parametrów elementu i uwzględnienie ich w obliczeniach.

Jedną z wad programów Analyser jest to, że na ekranie monitora w miejsce układu połączeń z naniesionymi wartościami elementów pojawia się lista elementów układu i węzły, między którymi są one rozmieszczone. Zmusza to operatora do żmudnego porównywania wartości i rozmieszczenia elementów układu z narysowanym na kartce schematem układu. Niedogodność ta wynika z założenia, że program będzie wykorzystywany na komputerach o małej pojemności pamięci operacyjnej, która wystarcza jedynie do realizacji jądra programu. Z tego powodu wydruk programu może okazać się niewystarczający do sporządzenia dokumentacji obliczeń, ponieważ - niezależnie od wersji programu - nie będzie zawierał schematu połączeń badanego układu.

Mimo tych mankamentów, oba oferowane przez Number One Systems Ltd programy wydają się być atrakcyjną propozycją dla użytkowników niewielkich systemów mikrokomputerowych, którzy zajmują się konstruowaniem układów elektronicznych.

Aleksander Weksler

Tabela 1. Porównanie Analyser I i Analyser II

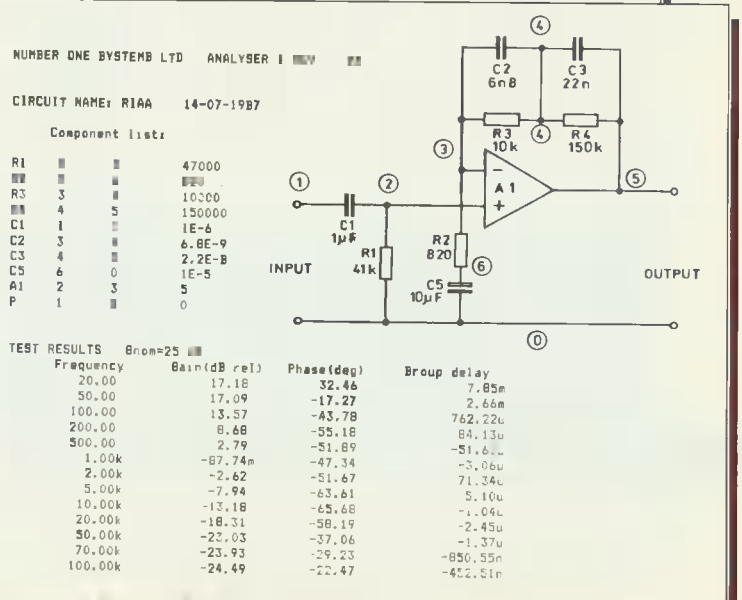
Nazwa Programu	Zakres analizowanych częstotliwości	Sposób przedstawiania wyników	Maksymalna liczba punktów charakterystyki	Inne udogodnienia
Analyser I	0,01Hz ... 1,1GHz	tabelarycznie	50	
Analyser II	0,01Hz ... 1,2GHz	tabelarycznie, graficznie	50	biblioteka elementów aktywnych i elementów w g oznaczeń fabrycznych

Tabela 2. Wykorzystanie programów Analyser I i II dla popularnych typów komputerów

Wersja programu	Typ komputera	Liczba węzłów	Liczba elementów	Cena (£)	Uwagi
Analyser I	Spectrum 48K	16	60	20	Nie oblicza opóźnienia grupowego
	Amstrad CPC	16		45	
	BBC	16	60	45	
	IBM PC	40	150	65	
Analyser II	BBC		60	130	biblioteka elementami aktywnymi
	IBM PC	60	180	195	biblioteka 26 elementami aktywnymi

Tabela 3 Modele obliczeniowe elementów analizowanego układu
- parametr przyporządkowywany elementowi przez program
- parametr wyliczany przez program
pozostałe parametry podawane przez operatora

Nazwa elementu	Parametry uwzględniane w programie	
	Analyser I	Analyser II
Rezystor	rezystancja R, pojemność równoległa 0.2 pF	
Indukcyjność, pojemność	traktowane jako elementy idealne	
Transformator	indukcyjność uzwojenia pierwotnego; przekładnia	
	rezystancja 1Ω połączona szeregowo z uzwojeniem pierwotnym	indukcyjność rozproszona uzwojenia pierwotnego
Tranzystor bipolarny (model hybryd π)	wzmocnienie prądowe h_{fe}, prąd emitera I_e, częstotliwość graniczna f_t, rezystancja wejściowa r'_{be}, pojemność baza-emiter C'_{be}	
	pojemności C_{cb} = 1 pF, C_{ce} = 1 pF	pojemności wyprowadzeń C_{ce}, C_{cb}, C_{be}
Tranzystor polowy	konduktancja przejściowa g_m	pojemności wyprowadzeń C_{gs}, C_{gd}, C_{ds}
Wzmacniacz operacyjny	różnicowa rezystancja wejściowa R_i = 10 GΩ, rezystancja wejściowa przy otwartej pętli R_o = 100 Ω, pasmo wzmocnienia jednostkowego f_u = 1 MHz, pojemność wspólna wejść C_{in} = 2 pF, pojemność różnicowa wejść C_{in} = 0.2 pF	
		wzmocn. z otwartą pętlą A_u



Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

(7) MS-DOS OD ŚRODKA

W kolejnym odcinku cyklu „MS-DOS" od środka omawiamy dalsze wywołania funkcji obsługi zbiorów i urządzeń zewnętrznych oraz wywołania funkcji zarządzających programami.

Funkcje obsługi zbiorów i urządzeń zewnętrznych.

Funkcja 45H otwiera drugi kanał logiczny (handle) umożliwiający dostęp do otwartego wcześniej urządzenia lub zbioru. Wszystkie operacje wykonywane w jednym kanale będą automatycznie odwzorowane w drugim. Nowy kanał nie umożliwia więc niezależnego manipulowania zbiorem czy urządzeniem. W szczególności w przypadku zbioru, wskaźnik odczytu/zapisu w obu kanałach jest zawsze identyczny! Podczas wywołania funkcji w rejstrze BX należy umieścić numer kanału, który został przyporządkowany zbiorowi (urządzeniu) przy jego otwieraniu. Jeśli nie wystąpi żaden błąd, to system otworzy drugi kanał i poda jego numer w rejestrze AX. W przeciwnym wypadku mogą pojawić się dwa błędy: 4 - zbyt dużo otwartych zbiorów oraz 6 - zły numer kanału (kanał o podanym w BX numerze nie istnieje lub nie został otwarty).

Funkcja 46H także otwiera drugi kanał logiczny, tym razem jednak numer drugiego kanału ustalany jest przez użytkownika, a nie przez system. Dzięki temu funkcja może być stosowana do redefinicji przypisania kanału. Użytkownik wywołując funkcje podaje numer pierwszego kanału w rejestrze BX i numer drugiego kanału w CX. Jeżeli kanał wybrany w rejestrze CX jest już wykorzystywany dla innego zbioru (urządzenia), to najpierw następuje zamknięcie tego zbioru. Możliwe błędy pojawiające się przy wywołaniu to: 4 - zbyt dużo otwartych zbiorów, 6 - zły numer kanału. Funkcję 46H można wykorzystać do zmiany standardowego przyporządkowania kanałów logicznych urządzeniom zewnętrznym. Przypuśćmy, że program wysyła dane na drukarkę (to znaczy do kanału logicznego 4). Chcąc spowodować wyprowadzenie ich do określonego zbioru należy najpierw otworzyć ten zbiór i uzyskać w BX numer przyporządkowanego zbiorowi kanału. Dla zachowania możliwości ponownego przypisania kanału 4 drukarce, należy otworzyć jeszcze jeden kanał jej przyporządkowany (funkcja 45H) i zapamiętać jego numer. Następnie do CX wpisać 4 i wywołać funkcję 46H, co spowoduje zamknięcie kanału 4 dla drukarki i otwarcie go dla nowego zbioru. Gdy dane w kanale 4 zechcemy ponownie wysyłać na drukarkę, wystarczy w rejestrze BX umieścić zapamiętany na wstępie numer, a w rejestrze CX - 4, i wywołać funkcję 46H.

Funkcja 47H podaje ścieżkę dostępu do wybranego katalogu w postaci ciągu ASCIIZ (znaki ASCII zakończone bajtem 00). Nie jest podawany symbol wybranego napędu dyskowego, ani też znak początkowy \ (backslash). Oznacza to, że gdy wybrany jest akurat katalog główny, funkcja poda pusty ciąg znaków. Rejestry DS:SI powinny zawierać adres przynajmniej 64-bajtowego obszaru pamięci, w którym system umieści ciąg ASCIIZ, zaś rejestr DL - liczbę kodującą napęd dyskowy (0 - napęd aktualny, 1 - napęd A:, itd.). Może wystąpić tylko jeden błąd - 0FH (zły numer napędu w DL). Dwie kolejne funkcje umożliwiają przeszukiwanie katalogu.

Funkcja 4EH służy do poszukiwania w katalogu pierwszego zbioru o podanej nazwie (może ona zawierać symbole wieloznaczne * i ?). Rejestry DS:DX powinny wskazywać na ciąg ASCIIZ określający ścieżkę dostępu do zbioru. Rejestr CL powinien specyfikować atrybuty poszukiwanego zbioru. Jeśli zbiór zostanie znaleziony, system wpisuje do aktualnego obszaru DTA 43 bajty zawierające informacje o zbiorze (tab. 1). Znajduje się tam między innymi ciąg ASCIIZ będący nazwą zbioru wraz z rozszerzeniem i oddzielającą je kropką. Jeśli rozszerzenia brak, kropka nie występuje. Podanie w rejestrze CX dowolnej kombinacji bitów, określających zbiór jako **hidden**, **system** lub **directory**, sprawia, że poszukiwane będą zarówno zbiory o podanych atrybutach jak i atrybucie **normal**. Jeśli podamy atrybut: **volume label**, poszukiwana będzie tylko pozycja będąca etykietą. Atrybuty **archiv** i **read only** są ignorowane. Atrybuty **volume label**, **directory** i **read only** nie są analizowane w wersjach systemu starszych od DOS 2.00.Możliwe są dwa błędy: 2 - nie znaleziono zbioru (podano złą ścieżkę dostępu); 12H - brak dalszych zbiorów (nie znaleziono zbioru o podanej nazwie).
Uwaga: w przypadku błędu, wskaźnik przeniesienia nie jest ustawiany. O błędzie świadczy niezerowa zawartość rejestru AX.

Funkcja 4FH kontynuuje poszukiwanie zbiorów o podanej (wieloznacznej) nazwie. Korzysta z informacji wpisanej do obszaru DTA przez funkcję 4EH (która musi zostać wcześniej wywołana) i umieszcza w tym obszarze informacje o zbiorze identyczne, jak funkcja 4EH (tab. 1). Jedyny możliwy błąd to 18 - nie znaleziono dalszych zbiorów.
Uwaga: podobnie jak przy poprzedniej funkcji, wskaźnik przeniesienia nie sygnalizuje błędu.

Funkcja 56H umożliwia zmianę nazwy zbioru. Rejestry DS:DX powinny wskazywać na starą nazwę, a ES:DI - na nową. Obie nazwy należy podać w postaci ciągów ASCIIZ, określając również ścieżki dostępu (na tym samym dysku). Jeśli obie ścieżki określają zbiory w dwóch różnych katalogach, to zbiór zostanie przepisany ze starego katalogu do nowego. Mogą wystąpić błędy: 2 - nie znaleziono zbioru, 3 - nie znaleziono ścieżki dostępu, 5 - dostęp niemożliwy i błąd 17 - podano różne napędy.
Na wydruku 1 przedstawiono program wykorzystujący funkcje rozszerzone do zmiany przyporządkowania urządzeń fizycznych kanałom logicznym.

Tabela 1. Informacje o znalezionym zbiorze podawane przez system w obszarze DTA po powrocie z wywołania funkcji 4EH

Przemieszczenie	Opis
00H...14H	obszar uzywany przez DOS do dalszych poszukiwań (za pomocą funkcji 4FH) - nie może być zmieniany przez użytkownika
15H	atrybuty znalezionego pliku
16H...17H	czas ostatniego dostępu
18H...19H	data ostatniego dostępu
1AH...1DH	wielkość zbioru (w bajtach; 1AH - najmniej znaczący bajt)

Tabela 2. Informacje zawarte w bloku sterującym wykorzystywanym przez funkcję 4BH przy AL=0

Przemieszczenie	Opis
00H...01H	adres segmentowy* ciągow ASCIIZ opisujących otoczenie (ang. enviroment string)
02H...05H	adres segmentowy wiersza poleceń
06H...09H	adres segmentowy pierwszego systemowego bloku FCB
0AH...0DH	adres segmentowy drugiego systemowego bloku FCB

* adres segmentowy oznacza 4 bajty (adres segmentu i przemieszczenie w ramach segmentu).

Tabela 3. Informacje zawarte w bloku sterującym wykorzystywanym przez funkcję 4BH przy AL=3

Przemieszczenie	Opis
00H...01H	adres segmentu, do którego ma być załadowany zbiór (z przemieszczeniem 0)
01H...02H	współczynnik relokacji (istotny tylko dla zbiorów typu EXE; najczęściej równy adresowi segmentu)

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Funkcje zarządzające programami

Funkcja 00H umożliwia zakończenie programu, to znaczy przekazanie sterowania do programu wywołującego. Jest ona wywoływana bezpośrednio (INT 21H z AX=0) lub przez przerwanie programowe 20H. Wykorzystanie funkcji 00H nie jest jednak najlepszym sposobem kończenia programu (przynajmniej w przypadku programów typu .EXE), gdyż trzeba spełnić szereg warunków - m.in. rejestrze CS musi być podany adres segmentu (z przesunięciem 0) zawierającego PSP (porównaj - MS-DOS od środka (2), mikroklan 6/87). Lepiej jest kończyć program przy użyciu funkcji 4CH, gdyż po wykonaniu funkcji 00H wektory przerwań 22H, 23H i 24H ustawiane są zgodnie z adresami zapamiętanymi w obszarze PSP (przesunięcie 0AH do 15H). A zatem, nawet jeśli zakończony program zmienił te wektory. funkcja ta przywróci ich wartości początkowe.

Funkcja 4BH pozwala załadować program lub podprogram do pamięci i ewentualnie go wykonać. Para rejestrów DS:DX powinna wskazywać na ciąg ASCIIZ, opisujący ścieżkę dostępu do pliku zawierającego program, zaś para rejestrów ES:BX - na blok parametrów opisujących sposób ładowania programu (tab. 2 i 3). Jeśli przy wywołaniu funkcji AL=0, to blok wskazywany przez ES:BX składa się z 14 bajtów. Po załadowaniu podprogramu, w czasie jego wykonywania, kanały logiczne (ang. handle), otwarte przez program wywołujący, są dostępne dla podprogramu wywoływanego. Program wywołujący może zmienić przyporządkowanie kanałów logicznych urządzeniom fizycznym i w ten sposób wpływać na sposób wykonania wywoływanego podprogramu.
Przed użyciem tej funkcji warto upewnić się, czy w systemie jest dostateczny obszar wolnej pamięci (funkcja 4AH).
Do programów, które można załadować i wykonać, należy m.in. interpreter poleceń DOS. Możemy mu wskazać ciąg znaków będących poleceniem do wykonania. Z kolei to polecenie może powodować wykonanie pliku zawierającego polecenia (ang. batch file), z których ostatnim powinno być polecenie EXIT, powodujące zakończenie wykonywania interpretera poleceń. W tym momencie kontrolę przejmuje na powrót program wywołujący. Jeśli AL=3, to nie następuje automatyczne przejście do wykonywania programu i nie jest tworzony nagłówek PSP. W ten sposób najczęściej ładuje się nakładki programowe. Jest to również prosty i efektywny sposób ładowania danych do pamięci.
Uwaga: Funkcja 4BH niszczy zawartość większości rejestrów, łącznie z rejestrami stosowymi - SS i SP. Aby zachować wartość rejestrów po załadowaniu i wykonaniu programu, należy więc zapamiętać je w pamięci dostępnej tylko dla programu wywołującego (np. w pamięci przechowującej kod tego programu), a po zakończeniu programu odzyskać je. Użycie do tego celu stosu nie jest możliwe. Mogą wystąpić następujące błędy: 1 - zły numer funkcji, 2 - nie znaleziono zbioru, 5 - dostęp niedozwolony, 8 - zbyt mało pamięci, 10 - złe otoczenie i 11 - zły format.

Funkcja 4CH umożliwia w prosty sposób zakończenie programu i przekazanie kodu zwrotnego (return code) - czyli wartości, jaka przy wyjściu z programu znajduje się w rejestrze AL. Jeśli program był wywołany z pliku zawierającego polecenia (ang. batch file), wartość kodu zwrotnego można testować za pomocą opcji DOS ERROR LEVEL.
Przy zakończeniu programu automatycznie zamykane są wszystkie zbiory otwarte za pomocą funkcji 3DF lub 3EH. Podobnie jak w przypadku funkcji 00H, przywracane są wektory przerwań 22H, 23H i 24H.

Funkcja 4DH umożliwia jednorazowe sprawdzenie kodu zwrotnego i sposobu zakończenia programu, (który przedtem załadowano i wykonano za pomocą funkcji 4BH). System podaje w AL kod zwrotny wysłany przez program, zaś w AH - sposób zakończenia. AH=0 oznacza zakończenie normalne, AH=1 - zakończenie spowodowane wciśnięciem CTRL-BREAK, AH=2 - zakończenie z powodu krytycznego błędu urządzenia, zaś AH=3 - zakończenie poprzez wywołanie funkcji 31H (zakończenie programu i nie usuwanie go z pamięci).

opr. Zbigniew Pojmański

```
NAME    ATTRIB
; Za pomocą polecenia ATTRIB można zmieniać (ustawiać lub zerować)
; atrybuty. Zbiór może być niewidoczny (Hidden), systemowy (System),
; tylko do odczytu (Read only), jako już archiwizowany
; (Archiv). Nazwy wszystkich zbiorów, których atrybuty ulegają
; zmianie, są wyprowadzane na ekran albo (w przypadku użycia /P na
; drukarkę. Jeśli operator wprowadzi polecenie w złej postaci,
; na konsolę zostanie wysłany komunikat o błędzie oraz wzór pra-
; widłowej postaci polecenia.
STACK   SEGMENT STACK
        DW      20 DUP(?)       ;definicja segmentu stosu
STACK   ENDS

CODE    SEGMENT 'CODE'
ASSUME  CS:CODE, DS:CODE, ES:NOTHING, SS:STACK
BELL            EQU     07H     ;kod ASCII sygnału akustycznego
CHANGE_MODE     EQU     43H     ;DOS: odczyt/zapis atrybutów
CON_OUT_HANDLE  EQU     01H     ;zdefiniowany początkowo przez DOS
                                ;kanał wyprowadzania na konsolę
CR              EQU     0DH     ;kod ASCII
DOS             EQU     21H     ;przerwanie dla wywoływania funkcji
DUPLICATE       EQU     45H     ;DOS: włączenie drugiego kanału
FALSE           EQU     00H     ;stała logiczna
FIND_FIRST      EQU     4EH     ;DOS: poszukiwanie pierwszego zbioru
FIND_NEXT       EQU     4FH     ;DOS: poszukiwanie kolejnego zbioru
FORCE_HANDLE    EQU     46H     ;DOS: przełączenie kanału
GET_DTA         EQU     2FH     ;DOS: odczyt adresu DTA (buf. dyskowego)
LF              EQU     0AH     ;kod ASCII LINE FEED
PARSE           EQU     29H     ;DOS: szukanie nazwy zbioru(ów)
PRINTER_CHANNEL EQU     04H     ;zdefiniowany początkowo przez DOS
                                ;kanał wyprowadzania na drukarkę
SET_DTA         EQU     1AH     ;DOS: ustalenie adresu bufora DTA
TERMINATE       EQU     4CH     ;DOS: zakończenie programu
TRUE            EQU     0FFH    ;stała logiczna
WRITE_HANDLE    EQU     40H     ;DOS: wyprowadzenie ciągu znaków

HELP_TEXT       DB      CR,LF,'Źle wprowadzone dane!',BELL,CR,LF,LF
                DB      'Użycie ATTRIB:',CR,LF,LF
                DB      'ATTRIB file /A(rchiv)<+,-> /H(ide)<+,->
                DB      /R(ead only)<+,-> /S(ystem)<+,-> /P(rint)'
                DB      CR,LF
                DB      '       + oznacza: ustawić atrybut',CR,LF
                DB      '       - oznacza: wyzerować atrybut',CR,LF,LF,0
COMMANDS        DB      'A','H','P','R','S',0   ;lista poleceń
COMMAND_MASK    DB      00100000B,00000010B,TRUE,00000001B,00000100B,0
  ;maski definujące (za wyjątkiem /P) bity określ poszczególne atrybuty
NEW_LINE        DB      CR,LF           ;ciąg 'RETURN, LINE FEED
ATTRIBUTE_MASK1 DB      FALSE           ;znacznik 'atrybut ma być ustawiony
ATTRIBUTE_MASK2 DB      FALSE           ;znacznik 'atrybut ma być wyzerowany'
COMMAND_BUFFER  DW      128 DUP(?)      ;kopia wprowadzonego wiersza polecenia
COMMAND_LENGTH  DB      ?               ;długość wiersza polecenia
DUMMY_FCB       DB      37 DUP(?)       ;funkcja PARSE wymaga istnienia FCB
                                        ;(który jednak nie jest tu używany)
FILE_LENGTH     DB      ?               ;długość nazwy znalezionego zbioru
FILE_NAME       DB      13 DUP(?)       ;bufor na nazwę zbioru
OUT_HANDLE      DW      ?               ;komórka na zapamiętanie numeru kanału
                                        ;związanego z wyprowadzaniem
PRINT_FLAG      DB      FALSE           ;znacznik określający miejsce wyprowa-
                                        ;dzania (FALSE -> CON:, TRUE -> PRN:
SECOND_HANDLE   DW      ?               ;miejsce na zapamiętanie numeru kanału
                                        ;związanego z konsolą - umożliwia przy-
                                        ;wrócenie drukowania na drukarce

START:  MOV     AH,GET_DTA              ;DOS: pobranie adresu bufora dyskowego
                                        ;(znajduje się tam wiersz polecenia)
        INT     DOS
        MOV     AX,ES                   ;ustalenie zdresu cryg. wiersza polec.
        MOV     DS,AX
        MOV     SI,BX
        MOV     AX,CS                   ;ustal. adresu kopii wiersza polecenia
        MOV     ES,AX
        MOV     DI,OFFSET COMMAND_BUFFER
        MOV     CX,SIZE COMMAND_BUFFER  ;liczba znaków do skopiowania
        REP     MOVSB                   ;skopiwanie polecenia do bufora
        MOV     AX,CS                   ;przestawienie segmentu danych (DS=CS)
        MOV     DS,AX                   ;(dane są teraz w tym samym segmencie)
        MOV     AX,CON_OUT_HANDLE       ;wpisanie jako kanału wyprowadzania
                                        ;standardowego wyjścia na konsolę
        MOV     OUT_HANDLE,AX
        XOR     CH,CH
        MOV     DI,OFFSET COMMAND_BUFFER
        MOV     CL,BYTE PTR [DI]
        MOV     COMMAND_LENGTH,CL       ;zapamiętanie długości polecenia
        TEST    CL,0FFH                 ;wprow. tekst po nazwie polecenia?
        CALL    OUTPUT_HELP             ;gdy nie, to wyprowadź HELP i kończ
        INC     DI                      ;pierwszy bajt DTA zawiera liczbę
                                        ;wprowadzonych znaków
        CLD
        MOV     AL,                     ;przeskoczenie początkowych spacji
        REPE    SCASB
        INC     CX                      ;korekcja rejestrów
        DEC     DI
        PUSH    DI                      ;ochrona adresu pierwszego znaku
        MOV     COMMAND_LENGTH,CL       ;zapamiętanie długości ciągu znaków
        CMP     CL,0                    ;czy są jakieś znaki?
        CALL    OUTPUT_HELP             ;jeśli nie, to obsłuż błąd
        MOV     SI,DI                   ;jeśli są, to przyg. dane dla PARSE
        MOV     DI,OFFSET DUMMY_FCB
        MOV     DX,SI                   ;oblicz adres konc. wiersza z polec
        ADD     DX,CX
        MOV     AL,0                    ;tryb 0 funkcji PARSE
        MOV     AH,PARSE                ;DOS: analiza wiersza polecenia
        INT     DOS
        CMP     SI,DX                   ;czy były jakieś znaki?
        CALL    OUTPUT_HELP             ;gdy nie, to wyprowadź HELP i kończ
        CMP     BYTE PTR [DI+1],        ;zły numer napędu?
        CALL    OUTPUT_HELP             ;gdy nie, to wyprowadź HELP i kończ
        SUB     DX,SI                   ;oblicz liczbę pozostałych znaków
        MOV     CX,DX
        XCHG    COMMAND_LENGTH,CL       ;i zapamiętaj ją
        SUB     CL,COMMAND_LENGTH       ;oblicz długość nazwy zbioru
        MOV     FILE_LENGTH,CL
        POP     SI                      ;zał adres początkowy nazwy zbioru
        MOV     DI,OFFSET FILE_NAME     ;załaduj adres końcowy nazwy zbioru
        REP     MOVSB                   ;przekopiuj nazwę zbioru
        MOV     BYTE PTR [DI],00H       ;wpisz terminator nazwy zbioru
        MOV     DI,SI
        MOV     CL,COMMAND_LENGTH       ;analiza pozostałych znaków
        MOV     AL,'/'                  ;poszukiwanie '/'
        REPNE   SCASB
        CMP     CL,0                    ;znaleziono?
        CALL    OUTPUT_HELP             ;gdy nie, to wyprowadź HELP i kończ
        MOV     SI,DI                   ;SI = DI
        PUSH    DI                      ;zapamiętaj adres ciągu poleceń
        MOV     BX,CX
;W pozost. części prog. następuje usunięcie polecenia wszystkich spacji.
;Do BX wpisuje l. pozost. znaków. "Małe" litery zamieniane są na "duże"
```

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

```
LOOP:   LODSB                           ;pobranie znaku
        CMP     AL,                     ;spacja?
        JZ      BLANK
        CMP     AL, a                   ;nie, czy znak ASCII < 'a' ?
        JL      STORE
        CMP     AL,'z                   ;nie, czy znak ASCII > 'z' ?
        JG      STORE
        AND     AL,11D11111B            ;zamień "małą" literę na "dużą"
STORE:  STOSb                           ;nie spacja, dołącz znak do ciągu
        JMP     END LOOP
BLANK:  DEC     BX                      ;spacja, nie dołączaj do ciągu
END LOOP
        LOOP    LOOP                    ;pozostań w pętli, aż nie będzie
                                        ;więcej znaków (CX=0)
        MOV     BYTE PTR [DI]           ;zakończ ciąg znakiem /
        CMP     BX,0                    ;czy są dalsze polecenia?
        CALL    OUTPUT_HELP             ;gdy nie, to wyprowadź HELP i kończ
        INC     BX                      ;dolicz '/' do liczby znaków
        POP     DI                      ;odzyskaj początk. adres ciągu znaków
DECODE: MOV     AL,[DI]                 ;pobierz znak polecenia (A,R,H,P,S)
        PUSH    DI                      ;zapamiętaj adres znaku polecenia
        MOV     DI,OFFSET COMMANDS      ;adres ważnego polecenia
        MOV     CX,OFFSET END COMMANDS + 1
        SUB     CX,DI                   ;oblicz dług. tabeli poleceń + 1
        REPNE   SCASB                   ;przeszukaj tabelę poleceń
        CMP     CX,0                    ;znaleziono polecenie?
        CALL    OUTPUT HELP             ;gdy nie, to wyprowadź HELP i kończ
        POP     DI                      ;odtworzenie adresu polecenia
        MOV     SI OFFSET END_MASK      ;obliczenie adresu maski atrybutów
        SUB     [illegible] CX
        MOV     AL,[..I]                ;pobranie maski atrybutów
        CALL    SET MASK                ;zmienienie wskaźnika atrybutów
                                        ;albo  przełączenie  wy na drukarkę
        CMP     BX,                     ;czy koniec ciągu poleceń?
        JNZ     DECODE                  ;jeśli nie, to analizuj dalej
        CALL    REDIRECT IO             ;jeśli tak, to dla PRINT_FLAG = TRUE
                                        ;przełącz wyprowadzanie na drukarkę
        CALL    PROCESS FILES           ;przełącz strybuty
        CALL    RESTORE IO              ;przełącz wy na powrotem na konsolę
        XOR     AL,AL                   ;kod powrotu = 00H
        MOV     AH,TERMINATE            ;DOS: zakończ program
        INT     DOS

[illegible]PUT HELP     PROC    NEAR
;jeśli nie jest ustaw  znacznik zera, następuje powrót do proc. głównej
;w przeciwnym wyp. wyprowadzany jest tekst HELP i następuje zakoń. prog.
        JNZ     NO_TEXT
        MOV     BX,OUT_HANDLE
        MOV     CX,OFFSET END_HELP      ;obliczenie długości HELP_TEXT
        MOV     DX,OFFSET HELP_TEXT     ;DX zawiera adr. pocz. HELP_TEXT
        SUB     CX,DX
        MOV     AH,WRITE_HANDLE         ;DOS: wyprowadzenie ciągu znaków
        INT     DOS
        XOR     AL,AL                   ;kod powrotu = 00H
        MOV     AH,TERMINATE            ;DOS: zakończenie programu
        INT     DOS
NO_TEXT: RET                            ;powrót do programu głównego
OUTPUT_HELP     ENDP

SET_MASK        PROC    NEAR
;ustaw maskę(i) zgodnie z poleceniami
;/P ustawia PRINT_FLAG; + ustawia ATTRIBUTE_MASK1; - ustawia ATTRIBUTE_MASK2
        CMP     AL,TRUE                 ;czy polecenie /P ?
        JNZ     SET_ATTRIBUTE           ;jeśli nie, to ustaw atrybut
        MOV     PRINT_FLAG,AL           ;jeśli tak, to ustaw znacznik drukow
        MOV     BP,1                    ;odstęp następnego '/' od P
        JMP     COMMON_SET              ;analizuj dalsze znaki
SET_ATTRIBUTE:
        CMP     BYTE PTR [DI+1],'+      ;czy po literze polecenia jest +
        JNZ     CHECK_MINUS             ;nie - sprawdź czy minus
        OR      ATTRIBUTE_MASK1,AL      ;był '+', trzeba ustawić atrybut
        JMP     NEXT_COMMAND
CHECK_MINUS:
        CMP     BYTE PTR [DI+1], -      ;czy po literze polecenia jest -
        JZ      RESET ATTRIBUTE         ;był '-', trzeba wyzerować atrybut
OUT_HELP: TEST  AL,0                    ;ani +, ani -, więc wyprowadź
        CALL    OUTPUT_HELP             ;HELP i wróć do programu głównego
RESET ATTRIBUTE:
        OR      ATTRIBUTE MASK2,AL      ;wyzerowanie strybutu
NEXT_COMMAND:
        MOV     BP,2                    ;odstęp następnego [illegible] od polecenia
COMMON SET:
        CMP     BYTE PTR DS:[DI+BP]
        JNZ     OUT_HELP                ;jeśli kolejne znaki nie zaczynają
        INC     BP                      ;się od '/', to wyprow. tekst HELP
        ADD     DI,BP                   ;adres następnego polecenia
        SUB     BX,BP                   ;liczba pozostałych znaków
        RET
SET MASK        ENDP
```

```
REDIRECT IO     PROC    NEAR
        ;przełączenie kanału wyprowadzania na drukarkę
        CMP     PRINT FLAG,TRUE         ;czy wyprowadzanie na drukarkę
        JNZ     END REDIRECT            ;nie - to wróć
        MOV     AH,DUPLICATE            ;DOS:  twórz drugi kanał wyprowadz
        MOV     BX,OUT HANDLE
        INT     DOS
        JC      ERROR HANDLE            ;jeśli błąd - to obsłuż
        MOV     SECOND_HANDLE,AX        ;zapamiętaj numer drug  kan  wypr
        MOV     AH,FORCE HANDLE         ;DOS: wymuszenie przełączenia
        MOV     BX,PRINTER_CHANNEL      ;kanału wyprowadz. na drukarkę
        MOV     CX,OUT_HANDLE

        INT     DOS
        JNC     END_REDIRECT            ;jeśli bez błędu - to wróć
ERROR_HANDLE:
        CALL    PROCESS_ERROR           ;obsługa błędu
END_REDIRECT:
        RET
REDIRECT_IO     ENDP

RESTORE_IO      PROC    NEAR
        ;odtworzenie połączenia logicznego kanału wyprowadzania
        CMP     PRINT_FLAG,TRUE         ;czy wyprowadzanie na drukarkę?
        JNZ     END_RESTORE             ;nie - to wróć
        MOV     AH,FORCE_HANDLE         ;DOS: wymuszenie przełączenia
        MOV     BX,SECOND_HANDLE        ;kanału wyprowadzania na poprze-
        MOV     CX,OUT_HANDLE           ;dni związany z konsolą
        INT     DOS
        JNC     END_RESTORE             ;jeśli bez błędu - to wróć
        CALL    PROCESS_ERROR           ;jeśli błąd - to obsłuż
END_RESTORE: RET
RESTORE_IO      ENDP

PROCESS_ERROR   PROC    NEAR
    ;obsługa błędów ogranicza się do przekazania w AL kodu błędu
        MOV     AH,TERMINATE            ;DOS: zakończenie programu
        INT     DOS
PROCESS_ERROR   ENDP

PROCESS_FILES   PROC    NEAR
        MOV     AH,SET_DTA          ;DOS: przełączenie bufora dyskowego
        MOV     DX,OFFSET COMMAND_BUFFER
        INT     DOS
        MOV     DX,OFFSET FILE_NAME     ;ustalenie adresu nazwy zbioru
        MOV     CX,DOIDO111B            ;atrybuty dla funkcji poszukującej
        MOV     AH,FIND_FIRST           ;DOS: poszukiw. pierwszego zbioru
        INT     DOS
        JC      FINO_ERROR              ;jeśli był błąd, to go obsłuż
FIND_LOOP:                              ;pętla poszukiwania zbiorów
        MOV     DX,OFFSET COMMAND_BUFFER + 1EH  ;adres znalezionej nazwy zb.
        MOV     AL,00H                  ;funkcja: odczytanie atrybutów
        MOV     AH,CHANGE_MODE          ;DOS: odczytanie/zapisanie atrybutów
        INT     DOS
        JC      CHANGE_ERROR            ;jeśli błąd, to go obsłuż
        MOV     AL,ATTRIBUTE_MASK2      ;zanegуj maskę zerowania atrybutu
        NOT     AL
        AND     CL,AL
        OR      CL,ATTRIBUTE_MASK1      ;ustawienie odpowiedniego atrybutu
        MOV     AL,1                    ;funkcja: zapisanie atrybutów
        MOV     AH,CHANGE_MODE          ;DOS: odczytanie/zapisanie atrybutów
        INT     DOS
        JC      CHANGE_ERROR            ;jeśli błąd, to [illegible] obsłuż
        MOV     DI,OFFSET COMMAND_BUFFER + 1EH
        MOV     DX,DI                   ;ustalenie adresu nazwy zbioru
        MOV     AL,0
        MOV     CX,13                   ;maksymalna długość nazwy zbioru
        REPNE   SCASB                   ;poszukiwanie końca nazwy zbioru
        DEC     DI
        SUB     DI,DX
        MOV     CX,DI
        MOV     BX,OUT_HANDLE           ;wybranie kan. wypr. nazwy zbioru
        MOV     [illegible],WRITE_HANDLE    ;DOS: wyprowadzenie ciągu znaków
        INT     DOS
        MOV     DX,OFFSET NEW_LINE      ;wyprowadzenie CR,LF
        MOV     CX,2
        MOV     BX,OUT_HANDLE
        MOV     [illegible],WRITE_HANDLE    ;DOS: wyprowadzenie ciągu znaków
        INT     DOS
        MOV     AH,FIND_NEXT            ;DOS: poszukiw. następnego zbioru
        INT     DOS
        JNC     FIND_LOOP               ;obsłuż znaleziony zbiór
        JMP     END_SET                 ;jeśli błąd to wróć
FIND_ERROR:
CHANGE_ERROR:
        CALL    PROCESS_ERROR           ;obsługa błędu
END_SET: RET
PROCESS_FILES   ENDP
CODE    ENDS
        END     START
```

mikrokram

Nowe drukarki laserowe

Drukarka laserowa LP-8GS firmy CPT Computation została opracowana specjalnie do zastosowań naukowo-technicznych. Wyposażana jest standardowo w 24 zestawy znaków, w tym 4 zawierające specjalne symbole matematyczne i techniczne. Pamięć RAM o pojemności 2,5 MB pozwala drukować, przy pełnej rozdzielczości, strony o wymiarach 8,5x14 cali, z rezerwą 852 KB pamięci na dodatkowe zestawy znaków. Maksymalna szybkość drukowania wynosi do 8 stron na minutę.
Jako wyposażenie dodatkowe przewidziano podajniki papieru maszynowego i kopert listowych. LP-8GS jest przystosowana do pracy w trybach HP LaserJet Plus, Epson FX-80, Diablo, Qume, Tektronix i ANSI. Cena 6 tys. USD.
Znacznie tańsza - tylko 1995 dolarów - jest drukarka firmy AST Research o nazwie Turbo Laser/EL. Jest ona przeznaczona dla tych, którzy nie przewidują zbyt wymyślnych zastosowań. Drukarka ta pracuje również z szybkością do 8 stron na minutę i ma rozdzielczość 300 dpi (punktów na cal) w trybie tekstowym oraz 150 dpi w trybie graficznym. Razem z drukarką zrealizowaną na mechanizmie Ricoh 4081 dostarczana jest karta sterownika, którą należy włożyć jako rozszerzenie do komputera typu IBM PC. Sterownik jest zbudowany w oparciu o mikroprocesor 68000 (12 MHz) i umożliwia emulację HP LaserJet i Diablo 630.
W wersji standardowej Turbo Laser/EL ma 512 KB RAM, pięć zestawów znaków i podajnik papieru formatu A-4. Producent oferuje również trzy zestawy rozszerzające. Pierwszy pozwala rozbudować pamięć RAM o 2,5 MB, zapewnia emulację dodatkowych typów drukarek i większy wybór zestawów znaków (995 USD). Drugi zestaw rozszerzający umożliwia powiększenie pamięci o dodatkowe 512 KB (395 USD). Wreszcie trzeci zestaw dodaje 3 MB RAM, 1 MB ROM i umożliwia interpretację języka PostScript (2495 USD). Byte 11/87 **(as)**

Byline firmy Ashton-Tate

Program jest próbą wejścia firmy Ashton-Tate na rynek systemów Desktop Publishing. Pracuje na komputerach IBM PC wyposażonych w minimum 384 KB RAM oraz kartę CGA, EGA lub Hercules. Akceptuje i przetwarza zbiory wytworzone za pomocą popularnych procesorów tekstu, elektronicznych formularzy rachunkowych, baz danych i programów graficznych. Byline nie wymaga żadnego dodatkowego wspomagania programowego czy sprzętowego. Zezwala na korzystanie ze wszystkich funkcji DOS bez opuszczania programu oraz tworzenie makrokomend. Opcje programu nie różnią się od możliwości oferowanych przez inne programy z dziedziny Desktop Publishing. Użytkownik dysponuje kilkunastoma krojami czcionek o rozmiarach od 6 do 72 punktów. Byline kosztuje 395 USD. Datamation **(an)**

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Zamiana LPT1 z LPT2

Użytkownicy komputerów PC, wyposażonych w dwie drukarki lub drukarkę i ploter z interfejsami równoległymi, ze zdziwieniem stwierdzają, że system operacyjny DOS nie zawiera poleceń, które umożliwiłyby skierowanie wydruku z programu użytkowego do urządzenia przyłączonego jako LPT2, ■ nie standardowo wybieranego LPT1. Nawet doświadczony użytkownik, który nie natknął się dotąd na ten problem, jest przekonany, że przełączenie takie może być wykonane poleceniem MODE. Tymczasem polecenie to pozwala przełączyć wyjście z interfejsu równoległego LPT1 na jeden z portów szeregowych COM1, COM2 (i dalsze), ale nie na inny interfejs równoległy.
Gdy wydruk ma zostać zrealizowany poleceniami systemu operacyjnego, rozwiązanie jest proste. Wystarczy w dyrektywie wskazać nazwę odpowiedniego urządzenia wyjściowego. Polecenia: TYPE DOKUMENT LPT2, COPY CON LPT2 lub DIR>LPT2 powodują wysłanie na drukarkę przyłączoną do LPT2 odpowiednio zawartości zbioru DOKUMENT, tekstu wprowadzonego z konsoli i katalogu zbiorów. Jeśli jednak wydruk ma nastąpić w programie, w którym nie przewidziano wyboru interfejsu drukarki, trzeba zastosować jeden z trzech poniższych sposobów.

Pierwszy z nich polega na zamianie w pamięci operacyjnej komputera adresów portów LPT1 i LPT2, drugi - na przechwyceniu strumienia wydruku kierowanego na LPT1 i skierowanie go na LPT2, trzeci - na modyfikacji programu użytkowego, z którego wprowadzany jest wydruk. Najprostszy w realizacji jest pierwszy z wymienionych sposobów.

Na karcie interfejsu równoległego umieszczone są pola połączeniowe, w których można, zgodnie ze wskazaniami producenta, zainstalować zwory, ustalając, czy dany blok pełni funkcję interfejsu urządzenia LPT1 czy LPT2. Konsekwencją dokonanego wyboru, jest przypisanie blokowi odpowiedniego adresu portu. Zwykle jest to adres 378H dla LPT1 i 278H dla LPT2. Adres ten jest zapisany w dwóch bajtach pamięci RAM dla LPT1 od adresu 408H, a dla LPT2 - od adresu 40AH. W pierwszym bajcie zapisane jest 8 mniej znaczących bitów adresu, ■ w drugim pozostałe bity. Przykładowo, w komórkach 408H i 409H znajdują się wartości 78H i 03H, co oznacza, że wydruk kierowany na LPT1 będzie odbywał się poprzez port o adresie 378H. Zawartość komórek 408H...40BH można sprawdzić wykorzystując program DEBUG (przykład 1).

Jeśli w komorkach 408H i 409H adres portu 378H zostanie zastąpiony przez 278H, wówczas wydruk kierowany na LPT1 będzie odbywać się poprzez port o adresie 278H, a więc przez urządzenie przyłączone do interfejsu LPT2.
Zmiany można wprowadzić posługując się programem DEBUG (przykład 2). Bardziej eleganckie i uniwersalne jest rozwiązanie, w którym realizowana jest zamiana zawartości komórek 408H i 409H z zawartością komórek 40AH i 40BH.
Najprostszy, jak można sądzić, program realizujący zamianę zawartości tych komórek został opublikowany w PC Magazine (9/87). Niestety, liczący pięć wierszy wydruk programu zawiera aż trzy błędy. Sposób utworzenia poprawnego programu PRNSWAP.COM podany został w przykładzie 3. Program narzędziowy DEBUG wykorzystany został tu jako Assembler. Każdorazowe wywołanie programu PRNSWAP powoduje zamianę adresów portów LPT1 i LPT2. Kod wynikowy programu zajmuje tylko 13 bajtów.
Czytelnicy, którzy nie mają doświadczenia z programem DEBUG mogą zastosować krótki program w języku Basic, który wykonuje identyczne operacje (przykład 4).
Adresy portów pozostają zamienione do czasu powtórnej inicjalizacji systemu (lub użycia programu modyfikującego te adresy). Wadą przedstawionego rozwiązania jest to, że po kilkakrotnym użyciu programu użytkownik traci orientację co do aktualnego stanu adresów. Z tego względu wygodniej jest zastosować program ZMIANA (przykład 5), który każdorazowo informuje o stanie przełączenia. Program ten jest nieco bardziej złożony i zajmuje więcej miejsca (63 bajty) niż PRNSWAP. Tym niemniej oba programy zajmują tyle samo miejsca na dyskietce. Wynika to z faktu, że MS-DOS rezerwuje dla każdego zapisywanego na dyskietce zbioru co najmniej dwa sektory (1024 bajty). Proces kompilacji przedstawiony jest w przykładzie 6, a komunikaty ukazujące się na ekranie przy kolejnych wywołaniach programu - w przykładzie 7.

Czytelnicy, którzy zainteresowani są innymi rozwiązaniami programowymi, znajdą w czasopiśmie BYTE z lipca 1987 r. (s. 18) opis rezydującego w pamięci programu LPT1XLPT2, który przechwytuje wydruk i kieruje go do urządzenia przyłączonego do interfejsu LPT2.

Romuald Szuniewicz

Przykład 1. Odczytanie adresów portów LPT1 i LPT2

```
Operacje                                        Komentarz
--------                                        ---------
A>DEBUG<Return>                                 wywołanie programu
-D 408 409<Return>                              zakres komórek do odczytania
0408  78 03 78 02                        x.x.   zawartość komórek
-Q<Return>                                      wyjście z programu
A>
```

Przykład 2. Zmiana adresu portu LPT1

```
Operacje                                        Komentarz
--------                                        ---------
A>DEBUG<Return>
-E 408<Return>                                  adres komórki do modyfikacji
0408 78.278<Return>                             modyfikacja zawartości
-Q<Return>
A>
```

Przykład 3. Utworzenie programu PRNSWAP.COM

```
Operacje                                        Komentarz
--------                                        ---------
A>DEBUG
-N PRNSWAP.COM<Return>                          nazwanie programu
-A<Return>                                      ustawienie trybu asemblera
XXXX:0100 POP DS<Return>                        kod programu
XXXX:0101 MOV AX,[408<Return>                     "
XXXX:0104 XCHG AX,[40A<Return>                    "
XXXX:0108 MOV [408],AX<Return>                    "
XXXX:010B INT 20<Return>                        koniec kodu
XXXX:010D<Return>                               zapis liczby bajtów
-R CX<Return>                                     w rejestrze CX
0D<Return>
-W<Return>                                      zapis pliku na dysku
Writing 000D bytes                              Komunikat o zapisie 13 bajtów
-Q<Return>
A
```

Przykład 4. Program ZAMIANA.BAS

```
10 DEF SEG=0
20 A1=PEEK(&H40A)            'ODCZYTAJ I ZAPAMIETAJ
30 A2=PEEK(&H40B)            ' ADRES PORTU LPT1
40 POKE &H40A,PEEK (&H408)  'WSTAW ADRES PORTU LPT2
50 POKE &H40B,PEEK (&H409)  ' W MIEJSCE ADRESU PORTU LPT1
60 POKE &H408,A1             'WSTAW ADRES PORTU LPT1
70 POKE &H409,A2             ' W MIEJSCE ADRESU PORTU LPT2
80 SYSTEM
```

Przykład 5. Program ZAMIANA.ASM

```
MAIN    SEGMENT
        ORG     100H
PORTAD1 EQU     408H    ; Komórka z adresem portu LPT1
PORTAD2 EQU     40AH    ; Komórka z adresem portu LPT2
PORTAD3 EQU     40CH    ; Komórka z adresem portu LPT3
START   PROC    FAR
        ASSUME  CS:MAIN,DS:MAIN
        PUSH    DS
        MOV     BX,0
        PUSH    BX
        MOV     DS,BX
        MOV     AX,[BX]+PORTAD1 ; wstaw adres LPT1 do AX
        XCHG    AX,[BX]+PORTAD2 ; wstaw adres LPT1 w miejsce LPT2
        MOV     [BX]+PORTAD1,AX ; wstaw adres LPT2 w miejsce LPT1
        MOV     CL,1
        SUB     CL,[BX]+PORTAD3 ; 0 jeśli stan normalny
                                ; 1 jeśli adresy zamienione
        MOV     [BX]+PORTAD3,CL   zapamiętaj wynik w PORTAD
        ADD     CL,49             zamień wynik na 1 lub 2
        MOV     AX,CX
        MOV     DS,AX
        MOV     NUMBER,CL         wstaw wynik do komunikatu
        MOV     DX,OFFSET MSG
        MOV     AH,9              wydrukuj komunikat
        INT     21H
        RET
MSG:    DB      'LPT1    LPT'    ;komunikat ekranowy
NUMBER  DB      0
        DB      13,10,'$'
        START   ENDP
MAIN    ENDS
        END     START
```

Przykład 6. Kompilacja i konsolidacja programu ZAMIANA

```
A>MASM ZAMIANA ,,

The Microsoft MACRO Assembler , Version 1.27
 (C) Copyright Microsoft Corp 1981, 1984

Warning Severe
Errors  Errors
0       0

A>LINK ZAMIANA ....

IBM Personal Computer Linker
Version 2.00 (C)Copyright IBM Corp 1981, 1982, 1983

Warning: No STACK segment

There was 1 error detected.

A>EXEBIN ZAMIANA ZAMIANA.COM
A>
```

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Przykład 7. Komunikaty przy kolejnym użyciu programu ZAMIANA.COM

```
A>ZAMIANA
LPT1==>LPT2
A>ZAMIANA
LPT1==>LPT1
A>
```

Wywołania z Turbo-Pascala

Istnieje kilka sposobów wywołania innego modułu programowego z programu napisanego w języku Turbo-Pascal. Pierwszy i chyba najczęściej używany, to stosowanie nakładek. Program jest dzielony przez programistę na kilka części, z których tylko jedna (aktualnie wykonywana) znajduje się w pamięci operacyjnej, pozostałe zaś przechowywane są na dysku. Fragment programu, zawarty w innej części niż dotąd wykonywana, może być użyty dopiero po przepisaniu go z dysku do pamięci operacyjnej (na miejsce poprzedniej części).
Kolejne dwie możliwości, to użycie wewnętrznych procedur języka Turbo-Pascal: CHAIN i EXECUTE. Za pomocą procedury EXECUTE wywoływany jest i wykonywany plik typu .COM, natomiast procedura CHAIN wymaga specjalnego formatu pliku wykonawczego (posiadającego rozszerzenie .CHN). Plik taki można utworzyć jedną z opcji kompilatora. Skompilowany w ten sposób program nie zawiera w sobie bibliotek czasu wykonania (ang. run time library) i korzysta z bibliotek programu głównego (tego, w którym znajduje się procedura CHAIN). Wymienione sposoby mają dużą wadę: żaden z nich nie daje możliwości wywołania modułu programowego skompilowanego innym translatorem (np. Microsoft Pascal). Ponadto CHAIN i EXECUTE nie pozwalają na powrót do modułu, z którego zostały wywołane.
System operacyjny DOS zawiera funkcję przeznaczoną do wywołania innego programu (4BH) load/execute program (ładuj i wykonaj program), ale jest ona obarczona poważnym błędem - podczas jej działania są niszczone zawartości z rejestrów wskazujących ostatnią pozycję stosu (SP wskaźnik stosu i SS segment stosu). Nie można więc w stosunku do tej funkcji stosować zwykłych wywołań za pomocą standardowych procedur MS-DOS i INT. Okazuje się jednak, że istnieje możliwość powrotu do modułu, z którego nastąpiło wywołanie. Nie jest to metoda elegancka i należy z niej korzystać tylko wtedy, gdy jest to konieczne.

Przygotowanie wywołania

Na wstępie należy zwolnić część pamięci operacyjnej potrzebnej na wywołany moduł. Programy utworzone w Turbo-Pascalu (jak i inne programy typu .COM) wywoływane z poziomu systemu operacyjnego, mają przydzieloną całą dostępną pamięć operacyjną. W dodatku kod programu i zmienne globalne są ulokowane w dolnym obszarze pamięci, natomiast stos jest umieszczony w górnym. Pozostała część pamięci operacyjnej jest przydzielna stercie (ang. heap). Nie ma więc wolnego fragmentu pamięci, w którym można by umieścić inny program.
Kompilator Turbo-Pascal ma jednak wbudowany mechanizm pozwalający na uzyskiwanie programów, które zwalniają część pamięci zaraz po wczytaniu ich do pamięci. Jest to możliwe tylko w przypadku kompilacji, której wynik ma zostać zapisany na dysku (Com-file). Opcje maximum free dynamic memory i minimum free dynamic memory, pozwalają ograniczyć do niezbędnego minimum rozmiary zajmowanej przez program pamięci. Należy przy tym pamiętać, że parametry przekazywane są do procedur przez stos, co ozanacza, że niezbędne jest co najmniej tyle wolnej pamięci, aby można było umieścić w stosie najobszerniejszą listę parametrów. Oszacowany rozmiar pamięci należy wprowadzić jako maksymalną i minimalną liczbę paragrafów (bloków po 16 bajtów) potrzebną dla stostu. Wartość ta powinna też zostać przypisana zmiennej (np. STACKSIZE - jak w przykładowym programie).
W przykładzie przyjęto rozmiar obszaru dynamicznego równy 4096 (co odpowiada 100H paragrafom) - wystarczający dla programów, które nie wykorzystują wskaźników lub struktur jako parametrów dla funkcji lub procedur.
W kolejnym kroku należy zwolnić cały obszar pamięci operacyjnej, która nie będzie wykorzystywana przez program główny. W tym celu można posłużyć się funkcją DOS-u 4AH(SETBLOCK). W rejestrze BX przekazywana jest liczba paragrafów, która ma być pozostawiona dla programu głównego, czyli nie ma być zwolniona. W rejestrze BX musi więc znaleźć się różnica adresów ostatniego segmentu pamięci zajmowanej przez program i ostatniego segmentu programu, powiększona o obszar zarezerwowany na stos (wartość STACKIZE):
BX:= Sseg–Cseg+STACKIZE;
Następnie za pomocą funkcji 4BH można wykonać program, który zmieści się w zwolnionym obszarze.

Oprac. Ryszard Rzędkowski
wg [illegible]2'87

```
program EXECDEMO; {Przykład wywołania [illegible] DOS "EXEC" z poziomu TURBO -PASCALA}

[illegible] STACKSIZE = [illegible] program potrzebuje 100H paragrafów [illegible] ; wartosc
                        ta musi być podana jako "Minimum" i "Maximum Stack Size"
                        przed kompilacją [illegible] postaci [illegible]

type
    REGISTER = record
                 AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : integer;
               end;

    [illegible] = string[80];

[illegible] REG      : REGISTER;

{[illegible]}

procedure SETBLOCK;

{ zwolnienie przez program obszaru pamieci [illegible] wykorzystywanej}

begin
with REG do
    begin
    [illegible] := Sseg-Cseg+STACKSIZE;    [illegible] wskazuje [illegible] koniec stosu, zwalniając wszystko ponad nie)
    ES := Cseg;                            (segment aktualnego kodu)
    [illegible] := $4A00;                  [illegible] przerwania 4AH DOS'a - SETBLOCK)
    MSDOS(REG);

    if odd [illegible]
        begin
        writeln ('SETBLOCK - ERROR ' [illegible] ='',AX);
        halt;
        end;
    end;
end;                [illegible]

{[illegible]}

procedure EXEC (PROGRAMNAME,COMMANDLINE : NAMESTRING);

{Oddaje sterowanie programowi PROGRAMNAME z parametrami wejściowymi COMMANDLINE ,
 z wywołaniem funkcji [illegible]

 type     PAR_TYPE = record
                ENV_SEG,COMM_OFF,COMM_SEG,
                D1,D2,D3,D4                    : integer;
          end;

 const    {Blok parametrow EXEC}
     PAR_BLOCK :    PAR_TYPE = (ENV_SEG : [illegible] ; COMM_OFF : [illegible] ; COMM_SEG : 0 ;
                                D1: 0 ; D2 : 0 ; D3 : 0 ; D4 : 0);

 { Definiuje [illegible] jako "Typed constants" i umieszcza [illegible] w segoencie kodu, [illegible]
   [illegible] dostep [illegible] nich z poziomu [illegible] }

   ASCIIZ :    [illegible] = '';
   SSSAVE :    integer    = 0;
   SPSAVE :    integer    = 0;
   [illegible] :    integer    = 0;      [illegible]
   FLAGSREG:   integer    = 0;      [illegible]

begin
ASCIIZ := PROGRAMNAME + chr(0);    {Utworz string typu ASCIIZ
COMMANDLINE := COMMANDLINE + chr(13);

with PAR_BLOCK do   {Zapisz COMMANDLINE do bloku parametrow}
    begin
    COMM_SEG := seg(COMMANDLINE);
    COMM_OFF := ofs(COMMANDLINE);
    end;
```

```
[illegible] funkcji EXEC w przypadku DOS'a 2.  moze byc problematyczne dlatego ten kod)

inline
    {najpierw zachowac rejestry DS,BP,SP,SS)
    ($1E/                   {PUSH    DS)
    $55/                    {PUSH    [illegible]
    $2E/                    {CS:       }
    $89/$26/SPSAVE/         {MOV     [SPSAVE],SP)
    $8C/$D0/                {MOV     AX,SS)
    $2E/                    {CS:       }
    $A3/SSSAVE/             'MOVE [SSAVE],AX)

    {przekopiuj CS [illegible] i ES,

    [illegible]/            {PUSH    [illegible]
    $0E/                    {PUSH    [illegible]
    $1F/                    {POP     [illegible]
    [illegible]/            {POP     [illegible]

    {Dla EXEC [illegible] musi wskazywac [illegible] string ASCIIZ zawierajacy nazwe programu wykonawczego
     dlatego DX bedzie [illegible] 2-giemu bajtowi ASCIIZ (bajt zawierajacy liczniki)}

    $BA/ASCIIZ+1/           {MOV     DX,ASCIIZ+1)
    $BB/PAR_BLOCK/          {MOV     BX,PAR_BLOCK)
    [illegible]             {MOV     AX,4B00 - kod funkcji EXEC)
    $CD/$21/                {INT     21      - wykonaj !

    {Po wykonaniu programu, najpierw trzeba odtworzyc zawartosci rejestrow)
    $2E/                    [illegible]        }
    $8B/$1E/SSSAVE/         {MOV     BX,[SSAVE])
    [illegible]             [illegible]     SS,BX)
    $2E/                    {CS:       }
    $8B/$26/SPSAVE/         {MOV     SP,[SPSAVE])
    $5D/                    {POP     BP)
    $1F/                    {POP     [illegible]

    { AX [illegible] rejestr wskaznikow [illegible] kody bledu EXEC, dlatego [illegible] je)
    $2E/                    {CS:       }
    [illegible]             [illegible]     [AXREG],AX)
    [illegible]             {PUSHF   , POP     [illegible]

    $2E/                    {CS:
     $A3/FLAGSREG);                 {MOV     [FLAGSREG],AX)
[----------]  [----------]
{ [illegible] '  [illegible]    DLA DOS'a 3.0 I KOD ZAWARTY MIEDZY LINIAMI PRZERYWANYMI

  [illegible] NASTEPUJACYM [illegible] :

  with REG do
        begin
        DS := seg(ASCIIZ);
        DX := ofs(ASCIIZ) +1 ;
        [illegible] := seg(PAR_BLOCK);
        [illegible] := ofs(PAR_BLOCK);
        [illegible] := [illegible]
        MsDos;
    end;              {WITH}
end;      {EXEC}

{[illegible]}
{[illegible]                                                    [illegible]}
{[illegible]     M A I N     P R O G R A M                      [illegible]}
{[illegible]                                                    [illegible]}
{[illegible]}
begin
SETBLOCK;       {zwolnienie pamieci}
writeln ('Calling Debugger [illegible] ...');
EXEC ('\DOS\DEBUG.COM','\AUTOEXEC.BAT');
writeln ('debugger returned [illegible] - program');
writeln (' [illegible] OF THIS TEST  '' [illegible]');
end.
```

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Klawiatura komputerów z rodziny PC XT/AT jest oddzielną fizycznie i logicznie jednostką wyposażoną we własny procesor i łącze szeregowe służące do komunikacji z płytą główną. Stosowanych jest kilka typów klawiatur o różnej liczbie klawiszy (zazwyczaj 83, 84 lub 101) i różnym ich rozmieszczeniu. Najczęściej spotykane są klawiatury 84-klawiszowe typu PCAT. Najnowsze klawiatury typu 101 dostosowane są głównie do komputerów PC AT. Zwiększono w nich liczbę klawiszy funkcyjnych oraz wyodrębniono grupę klawiszy sterowania kursorem.

KLAWIATURY

Sławomir Pietrowski

Klawisze

Typowa (84) klawiatura zawiera trzy grupy klawiszy. Centralnie usytuowana jest klawiatura standardowej maszyny do pisania. Z lewej strony wyodrębniono 10 klawiszy funkcyjnych, których znaczenie definiowane jest przez programy użytkowe. Z prawej strony znajduje się 18 klawiszy, których funkcje również mogą być definiowane programowo, lecz umieszczone na nich napisy sugerują typowy sposob wykorzystania (np. sterowanie kursorem).
Współpraca klawiatury z komputerem została zaplanowana w sposób zapewniający możliwość elastycznego dopasowywania funkcji poszczególnych klawiszy do potrzeb stosowanych programów. Do komputera przesyłane są kody klawiszy a nie standardowe kody przypisanych im znaków (tab. 1 i 2). Zamiana kodów przypisanych klawiszom na kody z tzw. rozszerzonego zestawu ASCII realizowana jest przez procedury BIOS-u.
Dla każdego klawisza przewidziany został osobny 8-bitowy kod (ang. scan code). Podsystem klawiatury wysyła inny kod, gdy klawisz jest naciśnięty i inny w momencie jego zwolnienia (ang. make/break scan code). W klawiaturze PC XT przy zwolnieniu klawisza kod tworzony jest przez dodanie wartości 80H do kodu wysyłanego w momencie naciśnięcia klawisza. Przykładowo przyciśnięcie klawisza ■ powoduje przesłanie kodu 01H, a jego zwolnienie - kodu 81H. W klawiaturze PC AT kod sygnalizujący zwolnienie klawisza przesyłany jest w postaci dwóch bajtów: kodu naciśnięcia poprzedzonego kodem F0H.

Gdy klawisz jest naciśnięty i trzymywany w tym położeniu kławiatura (z pewnym opóźnieniem) wysyła powtarzające się kody naciśnięcia aż do chwili zwolnienia klawisza. W przypadku jednoczesnego naciskania dwóch lub więcej klawiszy, tylko kod ostatnio naciśniętego jest przesyłany do komputera. Operacja powtarzania kończy się z chwilą zwolnienia tego klawisza. niezależnie od stanu pozostałych klawiszy. Jeśli połączenie z komputerem jest zablokowane, tylko pierwszy kod naciśnięcia jest przechowywany w buforze procesora klawiatury.

Interfejs klawiatury

Klawiatura połączona jest z systemem za pomocą zwiniętego spiralnie 4-żyłowego przewodu ekranowanego. Dwie z linii wykorzystywane są do doprowadzenia zasilania (+5VDC i masa). Pseudo-synchroniczna komunikacja szeregowa prowadzona jest liniami „zegara" (ang. clock) i „danych" (ang. data). Każda z linii sygnałowych jest sterowana przez bramki z otwartym kolektorem. Rozwiązanie umożliwia wymuszenie stanu logicznego „0", zarówno od strony klawiatury jak i komputera

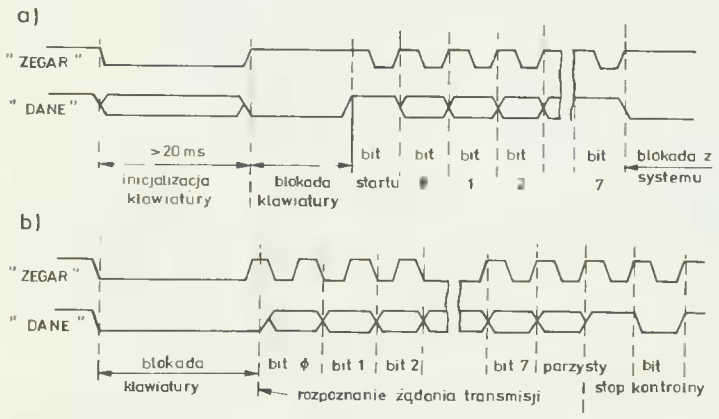
W stanie pasywnym, gdy dane nie są transmitowane, obie linie sygnałowe są w stanie logicznym „1". Transmisja między klawiaturą i systemem polega na przesyłaniu kolejnych bitów danej (linia „danych"), które wpisywane są do rejestru odbiorczego impulsami zegarowymi (linia „zegara").
Informacje przesyłane z klawiatury w systemie PC XT i PC AT mają inną postać (rys. 1).
W systemie PC XT jednostkę informacji stanowi paczka składająca się z bitu startu (poziom wysoki na linii „danych") i ośmiu bitów danych. (poczynając od najmniej znaczącego), kolejno synchronizowanych sygnałami zegara (opadające zbocze).
Klawiatura może rozpocząć nadawanie, gdy obie linie: „danych" i „zegara" utrzymywane są w stanie pasywnym 1.
Ustawienie przez komputer stanu „0" na linii „danych" wstrzymuje transmisję.
Utrzymanie stanu „0" na linii „zegara" przez okres ponad 200 ms powoduje przejście systemu klawiatury do inicjalizacji. W tym wypadku klawiatura potwierdza kodem AAH poprawność i gotowość działania.
W systemie PC AT dane i polecenia przesyłane są w obu kierunkach między klawiaturą i komputerem. Jednostka informacji składa się z 11 bitów: bitu startu („0"), ośmiu bitów danej (poczynając od najmniej znaczącego). bitu nieparzystości i bitu stopu („1"). Gdy dane są transmitowane do klawiatury komputer przesyła impulsy na linii „danych" i pozostawia niewysterowaną linię „zegara". Wysyłając dane do komputera lub odbierając je klawiatura generuje na linii zegara impulsy, które taktują przesyłanie kolejnych bitów. Komputer może wstrzymywać wysyłanie danych z klawiatury, wymuszając na linii zegara poziom niski („0"); stan linii „danych" jest wtedy nieważny.
Przed wysłaniem danej, klawiatura sprawdza stany linii sygnałowych:
- gdy linia „zegara" jest w stanie „0", to transmisja zostaje zablokowana i dana jest zapamiętywana w buforze;
- gdy linia „zegara" jest w stanie „1" i linia „danych" w stanie „0", to dana trafia do bufora, a klawiatura przechodzi w tryb odbioru informacji;
- gdy linie „zegara" i „danych" są w stanie „1", to klawiatura może rozpocząć transmisję do komputera.
Wartość transmitowanego bitu jest ustalana przed opadającym zboczem impulsu zegara i nie zmienia się aż do zbocza narastającego. Podczas transmisji, conajmniej raz na 60 ms klawiatura kontroluje stan linii „zegara".
Jeśli komputer wymusi poziom niski („0"), to wstrzywana jest dalsza transmisja.
Gdy kolizja nastąpi przed narastającym zboczem dziesiątego impulsu taktującego (gdy transmitowany jest bit nieparzystości), klawiatura przerywa sterowanie linii sygnałowych; w innym wypadku transmisja pełnego kodu musi zostać zakończona.
Komputer przed wysłaniem danej do klawiatury wymusza na linii „zegara" poziom niski przez okres ponad 60 ms. Transmisja jest inicjowana ustawieniem stanu „0" na linii danych. Równocześnie komputer przestaje sterować linię „zegara", co powoduje przejście tej linii w stan logiczny „1".
Gdy żądanie transmisji zostanie rozpoznane, klawiatura wysyła na linię zegara 11 impulsów taktujących przesyłanie danych. Po dziesiątym impulsie klawiatura ustawia na linii danych stan „0", taktowany jedenastym impulsem (bit stopu).
Taką reakcję komputer interpretuje jako odebranie danej przez klawiaturę i oczekuje na dane z klawiatury.
Każdemu przesłaniu danej do klawiatury musi towarzyszyć odpowiedź w czasie do 20 ms (o ile system nie zablokował łącza). Jeśli odpowiedź z klawiatury jest nieważna lub błędna, system ponownie wysyła polecenie lub daną.

Schemat klawiatury

W klawiaturach poszczególnych typów stosuje się różne rozwiązania konstrukcyjne. Przykładowy schemat (rys. 3) pokazuje typową koncepcję rozwiązania.
Centralnym elementem jest jednoukładowy procesor 8049 firmy Intel. Jego zadaniem jest:

1. Wykonanie autotestu (ang. basic assurance test) po włączeniu zasilania i na ewentualne żądanie systemu. Test sprawdza sumę kontrolną całej pamięci programu sterującego ROM, adresowanie i zwarcia między bitami pamięci RAM oraz wysterowuje sygnalizacyjne diody świecące. Test trwa od 600 do 900ms i kończy się wysłaniem do systemu kodu AAH, jako informacji o poprawnym działaniu klawiatury, lub kodu FCH w wypadku stwierdzenia błędów;
2. Sterowanie transmisją szeregową między klawiaturą i systemem za pomocą układu U1 (bramka z otwartym kolektorem);
3. Sterowanie protokołem wzajemnego potwierdzania (ang. handshake) danych przesyłanych łączem;
4. Przeglądanie klawiszy z wykorzystaniem układu U3 (74LS138) i buforowanie conajmniej szesnastu kodów w wewnętrznej pamięci procesora;
5. Sterowanie diodami świecącymi wskazującymi stan klawiatury lub komputera **Caps Lock**, **Num Lock** i **Scroll Lock** w PC AT lub **Power-On** w PC XT.

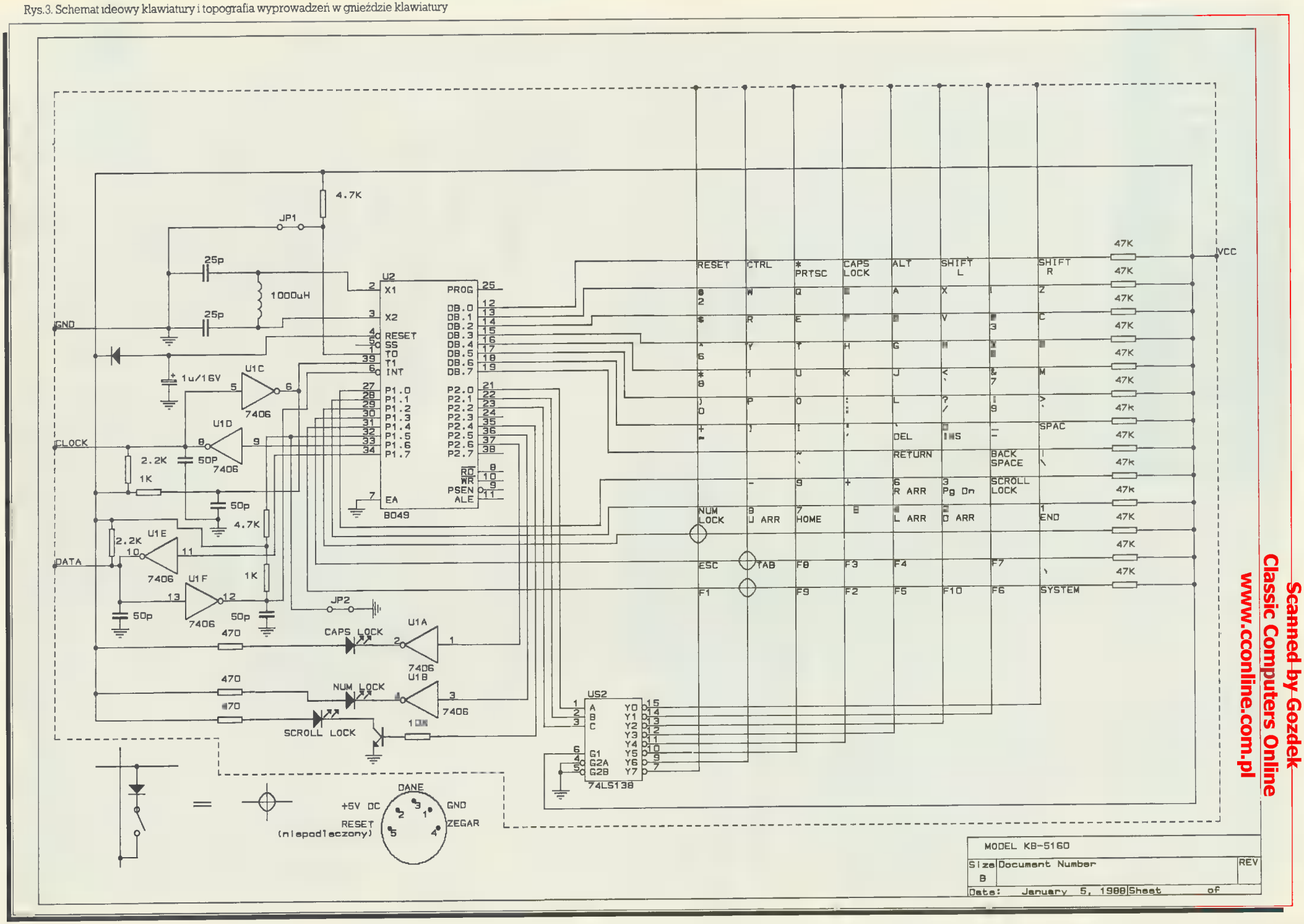
Konstrukcja układowa klawiatury jest stosunkowo prosta; cała mądrość spoczywa w treści programu zawartego w pamięci stałej procesora 8049.



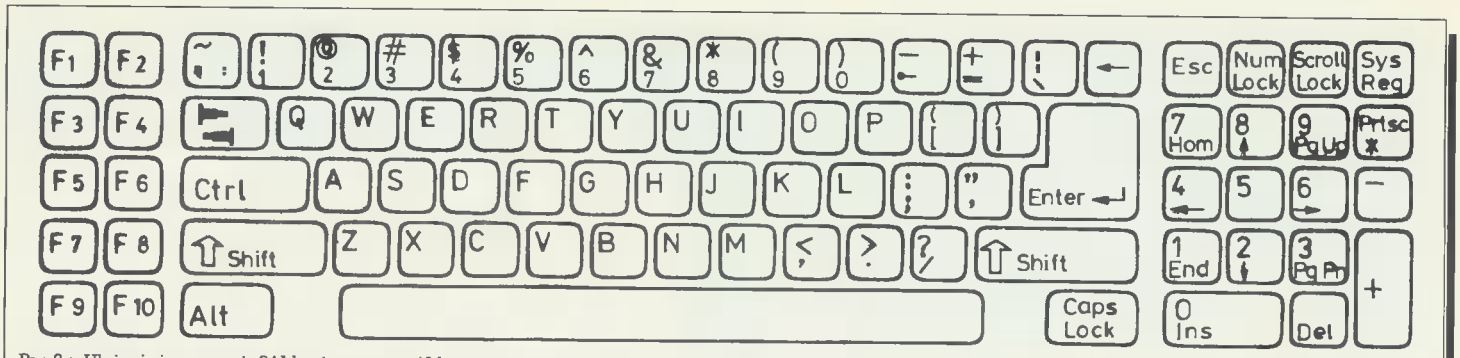
Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Rys. 1: a- transmisja danej z klawiatury do systemu PC XT,
b- transmisja danej systemu PC AT do klawiatury,
c- transmisja danej z klawiatury do systemu PC AT

Rys.3. Schemat ideowy klawiatury i topografia wyprowadzeń w gnieździe klawiatury



Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl



Rys.2.a. Ułożenie i numeracja 84 klawiszy typowej klawiatury

3B F1	3C F2	29 A9	02 82	03 83	04 84	05 85	06 86	07 87	08 88	09 89	0A 8A	0B 8B	0C 8C	0D 8D	2B AB	0E 8E		01 81	45 C5	46 C6	••
3D BD	3E BE	0F 8F	10 90	11 91	12 92	13 93	14 94	15 95	16 96	17 97	18 98	19 99	1A 9A	1B 9B		▪		47 C7	48 C8	49 C9	37 B7
3F BF	40 C0	1D 9D		1E 9E	1F 9F	20 A0	21 A1	22 A2	23 A3	24 A4	25 A5	26 A6	27 A7	28 A8		1C 9C		4B CB	4C CC	4D CD	4A CA
41 C1	42 C2	2A AA		2C AC	2D AD	2E AE	2F AF	30 B0	31 B1	32 B2	33 B3	34 B4	35 B5		36 B6			4F CF	50 D0	51 D1	
43 C3	44 C4	38 B8							39 B9							3A BA		52 D2		53 D3	4E CE

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Rys.2.b. Kody klawiszy klawiatury PC XT

05	06	0E	16	1E	26	25	2E	36	3D	3E	46	45	4E	55	5D	66		76	77	7E	84
04	0C	0D	15	1D	24	2D	2C	35	3C	43	44	4D	54	5B		▪		6C	75	7D	7C
03	0B	14		1C	1B	23	2B	34	33	3B	42	4B	4C	52		5A		6B	73	74	7B
83	0A	12		1A	22	21	2A	32	31	3A	41	49	4A		59			69	72	74	
01	09	11							29							58		70		71	79

Rys.2.c. Kody klawiszy klawiatury PC AT

Komunikacja programowa klawiatury i systemu PC AT

System PC AT może wysyłać różne polecenia zmieniające tryb pracy klawiatury:
- **zerowanie** (ang. reset, kod FFH) - żądanie inicjalizacji i wewnętrznego testu klawiatury. Klawiatura przesyła potwierdzenie ACK i oczekuje na reakcję, tzn. ustawienia linii „zegara" i „danych" w stan „1" na okres przekraczający 500 mikrosekund. Efekt działania tego rozkazu jest zbliżony do zerowania po włączeniu zasilania;
- **powtórz daną** (ang. resend, kod FEH) - żądanie powtórzenia ostatnio wysłanej, prawdopodobnie błędnej danej;
- **nic nie rób** (ang. no-operation NOP, kody od FDH do F7H) - rozkaz bez znaczenia dla klawiatury;
- **ustaw standardowe wartości** (ang. set default, kod F6H) - ustawienie standardowych parametrów klawiatury - takich jak w stanie początkowym. Klawiatura odpowiada znakiem ACK, zeruje bufor wyjściowy, ustawia wartości początkowe i kontynuuje przeglądanie klawiszy;
- **przejdź w stan pasywny po ustawieniu wartości początkowych** (ang. default disable, kod F5H) - podobnie jak poprzednio, lecz przerywane jest przeglądanie klawiszy i klawiatura oczekuje na dalsze instrukcje;
- **przejdź w stan aktywny** (ang. enable, kod F4H) - klawiatura potwierdza przyjęcie rozkazu znakiem ACK, zeruje bufor wyjściowy i rozpoczyna przeglądanie klawiszy;
- **ustaw wartości opóźnienia i częstość repetycji** (ang. set typematic rate/delay, kod F3H) - klawiatura odpowiada znakiem ACK, wstrzymuje przeglądanie klawiszy i oczekuje na parametr (jeden bajt).

Poszczególnym bitom przesyłanego parametru przypisywane jest następujące znaczenie:
bit $b_7 = 0$;
bity b_6 i b_5 określają opóźnienie wg zależności -
opóźnienie $= (1+b_6b_5) * 250$ ms - 20%;
bity $b_4,\ldots,b_0$ określają częstość repetycji wg równania -
okres repetycji $= (8+b_2b_1b_0) * 2^{[b_4b_3]} * 0{,}00417$ s.
Częstość repetycji można dobrać od 2 do 30 powtórzeń wysłania kodu stale naciskanego klawisza na sekundę (częstość początkowa - 10 znaków/s). Opóźnienie początkowe (czas uznania klawisza za stale naciskany) może być zmieniane od 250ms do 1250ms; wartość początkowa wynosi 500ms - 20%
- **wyślij echo** (ang. echo, kod EEH) - reakcją klawiatury jest wysłanie do systemu znaku EEH.
- **ustaw wskaźniki trybu** (ang. set/reset mode indicators, kod EDH) - klawiatura odpowiada kodem ACK, wstrzymuje przeglądanie klawiszy i oczekuje bajtu parametru. Parametr zawiera informację o stanie wskaźników (zapaleniu diod świecących), odpowiadających klawiszom Caps Lock, Numeric Lock i Scroll Lock. Poszczególnym wskaźnikom są odpowiednio przyporządkowane bity b_2, b_1, b_0 bajtu parametru.

Klawiatura może również wysyłać szereg poleceń i odpowiedzi do systemu:

- **powtórz nadanie** (ang. resend, kod FEH) - żądanie powtórzenia błędnej transmisji;
- **potwierdzenie** (ACK, kod FAH) - potwierdzenie przyjęcia polecenia z komputera;
- **przepełnienie** (ang. overrun, kod 00H) - informacja o przepełnieniu bufora pamięci w klawiaturze; niewydolności komputera do odbioru kodów naciskanych klawiszy. Kod 00H umieszczany jest jako siedemnasty znak w buforze wyjściowym klawiatury i przesyłany w momencie zapełnienia bufora;
- **błąd diagnostyczny** (ang. diagnostic failure, kod FDH lub FCH) - wysyłany w momencie wykrycia błędów odczytu klawiszy. Jeśli błędy zdarzą się podczas testu podstawowego, klawiatura zawiesza przeglądanie klawiszy i czeka na rozkazy systemu lub inicjalizację od włączenia zasilania. W innym przypadku, nadal jest przeszukiwana matryca klawiszy;
- **zwolnienie klawisza** (ang. break code prefix, kod F0H) - poprzedza kod klawisza w momencie jego zwolnienia;
- **zakończenie testu podstawowego** (ang. basic assurance test completion code, kod AAH) - poprawne zakończenie pełnego testu klawiatury;
- **echo** (ang. echo, kod EEH) - odpowiedź na rozkaz echo z systemu.

Procedury obsługi klawiatury

Podstawowym zadaniem procedury obsługi klawiatury jest zamiana kodów klawiszy w znaki nazwane rozszerzeniem ASCII (ang. Extended ASCII) lub operacje. Danymi wyjściowymi są jednobajtowe kody znaków lub kody rozszerzone, bądź też funkcje wykonywane przez przerwania albo w procedurze obsługi klawiatury.

Procedura BIOS-u interpretująca kody klawiszy, wywoływana jest przez przerwanie 16H (INT 16), z poleceniem przekazywanym w rejestrze AH:

(AH)=0 - odczytaj kolejny znak ASCII wprowadzony z klawiatury; w rejestrze AL procedura przekazuje kod ASCII, zaś w rejestrze AH kod klawisza;

(AH)=1 - ustaw znacznik zera zależnie od tego czy znak ASCII jest gotowy do odczytania, jeśli (ZF)=1 znaku nie ma, gdy (ZF)=0, to znak znajduje się w rejestrze (AX), a w buforze pozostają stają wartości wejściowe do utworzenia znaku;

(AH)=2 - wprowadź do rejestru AL aktualny status klawiszy specjalnych (ang. shift flags); kolejno, od pozycji najbardziej znaczącej zapisane są stany klawiszy Ins, Caps Lock, Num Lock, Scroll Lock i zwolnienia klawiszy Alt, Ctrl, Left Shift, Right Shift.

Tabela 1 przedstawia kody znaków (ASCII) przekazywane przez BIOS systemowi operacyjnemu lub programom użytkowym. Kod „-1" oznacza, iż odpowiadająca mu kombinacja klawiszy jest przez procedurę ignorowana (bez odpowiedzi zwrotnej). Pewne funkcje nie mogą być odzwierciedlone w kodach ASCII. Stosuje się wtedy dwubajtowe wartości, tzw. kody rozszerzenia. Wartość 00H (Nul) przekazywana przez procedurę BIOS-u w rejestrze AL sygnalizuje konieczność odczytania drugiego bajtu z rejestru AH. Wartość drugiego bajtu jest zwykle kodem naciśniętego klawisza. Znaczenie kodów rozszerzenia obrazuje tabela 2. Warto zaznaczyć, iż w przypadku naciśnięcia kombinacji klawiszy Alt, Ctrl, Shift i sytuacji, gdy tylko jeden z nich może być uznany, kolejność priorytetów jest następująca: najpierw klawisz Alt, potem Ctrl i na końcu Shift.

Pewne kombinacje klawiszy powodują wykonanie nietypowych funkcji. Są to:

Zerowanie systemu (ang. system reset) wywołuje jednoczesne naciśnięcie klawiszy Alt, Ctrl, Del.

Przerwanie programowe (ang. break) o numerze 1AH wywołuje jednoczesne naciśnięcie klawiszy Ctrl i Break.

Zawieszenie (ang. pause) jest powodowane naciśnięciem klawiszy Ctrl i Num Lock. Procedura obsługi klawiatury realizuje pętlę oczekiwania na jakikolwiek kod za wyjątkiem Num Lock.

Wydrukowanie treści ekranu na drukarce (ang. print screen) wywołuje naciśnięcie klawiszy Shift i PrtSc.

Procedura obsługi klawiatury buforuje dane w 15-bajtowej kolejce typu FIFO. Bufor trudno przepełnić nawet biegłej maszynistce. Gdyby jednak przepełnienie nastąpiło, to klawisz zostanie zignorowany a użytkownik ostrzeżony sygnałem dźwiękowym (ang. bell). Procedura obsługi klawiatury chroni również przed repetycją (ang. typematic action) dla klawiszy: Ctrl, Shift, Alt, Num Lock, Scroll Lock, Caps Lock i Ins.

Uwagi eksploatacyjne

Wybierając klawiaturę, należy zwrócić uwagę na pewne cechy mechaniczne klawiszy i całej obudowy. Klawisz powinien działać w oparciu o efekt dodatniego sprzężenia zwrotnego, powstający przy jego naciskaniu (rys. 4a). Przykładanie narastającej siły powinno sprawiać wrażenie przełamania mechanicznego oporu klawisza przed elektrycznym załączeniem.

Charakterystyczne klawisze typu Enter i Shift powinny być znacznie większe, co ułatwia pracę operatora.

Opisy klawiszy na ich główkach powinny pozostawać możliwie długo czytelne i wyraziste. Jedną z możliwości rozwiązania problemu jest użycie techniki nakładania dwóch warstw plastiku o różnych kolorach (rys. 3b). Dobre klawisze powinny wytrzymać ponad 20 mln naciśnięć, bez uszczerbku dla ich funkcjonowania

Uwagi **Sławomira Piotrowskiego**
zebrała i opracowała **Elwira Kądzielska**

Interpretacja kodów klawiszy przez procedury BIOS-u

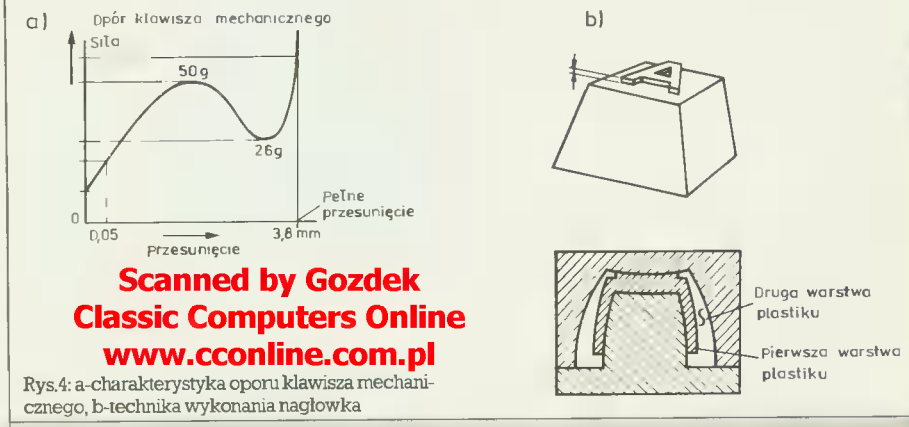
Numer klucza	Znaczenie podstawowe	z Shift	z Ctrl	z Alt
1	'	~	-1	-1
2	1	!	-1	[1]
3	2	@	Nul(000)[1]	[1]
4	3	#	-1	[1]
5	4	$	-1	[1]
6	5	%	-1	[1]
7	6	f	RS (030)	[1]
8	7	&	-1	[1]
9	8	*	-1	[1]
10	9	(	-1	[1]
11	10	)	-1	[1]
12	-	_	US(031)	[1]
13	=	+	-1	[1]
14	/	:	FS (028)	-1
15	BS(008)	BS(008)	Del (127)	-1
16	→(009)	←[1]	-1	-1
17	q	Q	DC1 (017)	[1]
18	w	W	ETB (023)	[1]
19	e	E	ENQ (005)	[1]
20	r	R	DC2 (018)	[1]
21	t	T	DC4 (020)	[1]
22	y	Y	EM (025)	[1]
23	u	U	NAK (021)	[1]
24	i	I	HT (009)	[1]
25	o	O	SI (015)	[1]
26	p	P	DLE (016)	[1]
27	[	{	Esc (027)	-1
28	]	}	GS (029)	-1
30 Ctrl	-1	-1	-1	-1
31	a	A	SOH (001)	[1]
32	s	S	DC3 (013)	[1]
33	d	D	EOT (004)	[1]
34	f	F	ACK (006)	[1]
35	g	G	BEL (007)	[1]
36	h	H	BS (008)	[1]
37	j	J	LF (010)	[1]
38	k	K	VT (011)	[1]
39	l	L	FF (012)	[1]
40	;	:	-1	-1
41	`	"	-1	-1
43 CR Enter	CR Enter	LF (010)	-1	
44 Shift lewy	-1	-1	-1	-1
46	z	Z	SUB (026)	[1]
47	x	X	CAN (024)	[1]
48	c	C	EXT (003)	[1]
49	v	V	SYN (022)	[1]
50	b	B	STX (002)	[1]
51	n	N	SO (014)	[1]
52	m	M	CR (013)	[1]
53	,	<	-1	-1
54	.	>	-1	-1
55	/	?	-1	-1
57 Shift prawy	-1	-1	-1	-1
58 Alt	-1	-1	-1	-1
61	SP	SP	SP	SP
64 Caps Lock	-1	-1	-1	-1
65	Nul[1]	Nul[1]	Nul[1]	Nul[1]
66	Nul[1]	Nul[1]	Nul[1]	Nul[1]
67	Nul[1]	Nul[1]	Nul[1]	Nul[1]
68	Nul[1]	Nul[1]	Nul[1]	Nul[1]
69	Nul[1]	Nul[1]	Nul[1]	Nul[1]
70	Nul[1]	Nul[1]	Nul[1]	Nul[1]
71	Nul[1]	Nul[1]	Nul[1]	Nul[1]
72	Nul[1]	Nul[1]	Nul[1]	Nul[1]
73	Nul[1]	Nul[1]	Nul[1]	Nul[1]
74	Nul[1]	Nul[1]	Nul[1]	Nul[1]
90	Esc	Esc	Esc	-1
95 Num Lock	-1	-1	Pause[2]	-1
100 Scroll	-1	-1	Break[2]	-1

Numer klucza	z Num Lock	Znaczenie podstawowe	z Alt	z Ctrl
91	7	Home[1]	-1	Czyszczenie ekranu
92	4	←[1]	-1	Cofnięcie o słowo[1]
93	1	End[1]	-1	Czyszczenie do końca linii[1]
96	8	Up [1]	-1	-1
97	5	-1	-1	-1
98	2	Down [1]	-1	-1
99	0	Ins	-1	-1
101	.	Page Up [1]	-1	na początek tekstu
102	6	→[1]	-1	słowo wprzód[1]
103	3	Page Down [1]	-1	Czyszczenie do konca ekranu[1]
104	.	Del[1],[2]	[2]	[2]
106	*	*	-1	Nul[1]
107	-	-	-1	-1
108	+	+	-1	-1
105	Sys Re	Bez funkcji		

[1]-kody rozszerzone [2]-funkcje specjalne

Tabela 2. Kody rozszerzone

Kod Dodatkowy	Funkcja
Znak Nul	
15	←
16-25	Alt Q, W, E, R, T, Y, U, I, O, P
30-38	Alt A, S, D, F, G, H, J, K, L
44-50	Alt Z, C, V, B, N, M
59-68	F1 do F10 kl. funkcyjne w trybie podst.
71	Home (lewy górny róg ekranu)
72	Up
73	Page Up i Home
75	←
77	→
79	End
80	Down
81	Page Down i Home
82	Ins (dopisywanie)
83	Del (kasowanie)
84-93	F11 do F20 (Shift F1 do F10)
94-103	F21 do F30 (Ctrl F1 do F10)
104-113	F31 do F40 (Alt F1 do F10)
114	Ctrl PrtSc (Start/stop kop. na drukarce)
115	Ctrl ← (cofnięcie o słowo)
116	Ctrl → (przesunięcie o słowo)
117	Ctrl End (kasowanie do końca linii)
118	Ctrl PgDn (kasow. do końca ekranu)
119	Ctrl Home (czyszczenie ekranu)
120-131	Alt 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, -, =(klawisze 2-13)
132	Ctrl PgUp (na początek tekstu)



Rys.4: a-charakterystyka oporu klawisza mechanicznego, b-technika wykonania nagłowka

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Sławomir Piotrowski zajmuje się konstrukcją, uruchamianiem i diagnostyką sprzętu komputerowego. Kontakt: Zakład Elektroniki ZEMIK tel. 106726 (Warszawa).

PROGRAMOWANIE STEROWNIKÓW W PASCALU (2)

W drugiej części artykułu o programowaniu systemów pracujących w czasie rzeczywistym przedstawiamy moduł MANAGER oraz program ilustrujący wykorzystanie koncepcji opisanej w poprzednim numerze mikroklanu.

Moduł MANAGER

Moduł MANAGER jest typowym modułem użytkowym, opracowanym całkowicie w języku Pascal/MT+, dołączanym do programu, tak samo jak standardowe biblioteki kompilatora Pascal/MT+. Zawiera operacje, które pozwalają konstruować algorytmy współbieżne, jak też wyrazić w kategoriach języka wysokiego poziomu oddziaływania między procesami.
Współpracę procesów można zapewnić poprzez współdzielone zmienne i sygnały. Współdzielone zmienne służą do przekazywania danych między procesami. Sygnały, które są eksportowane z modułu MANAGER, nie pozwalają na przekazywanie danych, lecz zapewniają synchronizację procesów.
Dla sygnałów są zdefiniowane dwie operacje: proces może wysyłać lub otrzymywać sygnał.
Każdy sygnał związany jest z pewnym warunkiem, dotyczącym wybranych zmiennych i jego wysłanie powinno być następstwem wystąpienia takiego wyróżnionego warunku.
Wysłanie sygnału uaktywnia najwyżej jeden proces, zaś wysłanie sygnału, na który nic nie czeka, jest rozumiane jako operacja pusta.
W module MANAGER zadeklarowano typ danych **SIGNAL**, które pełnią rolę sygnałów. Typ ten został ukryty przed użytkownikiem modułu i na zewnątrz jest zadeklarowany w następujący sposób:

```
TYPE
SIGNAL=WORD;
```

Natomiast wewnątrz modułu jest on typem wskaźnikowym do rekordu opisującego proces:

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

```
TYPE
SIGNAL =-DESCRPROC;
```

Taka podwójna deklaracja nie powoduje komplikacji ze względu na zgodność wymienionych typów, natomiast pozwala na uchronienie się przed skutkami nieautoryzowanego dostępu do opisu procesu przez użytkownika. Na sygnał może czekać jednocześnie kilka procesów i w takim wypadku tworzą one kolejkę, której początek wskazywany jest przez zmienną sygnałową.
Z tego opisu wynika, że procesy w programie mogą znajdować się w jednym z dwóch stanów: aktywnym (proces aktualnie realizowany) lub pasywnym. Proces, do którego przydzielono procesor jest procesem aktywnym. W systemie jednoprocesorowym w każdej chwili tylko jeden proces może być aktywny. Przydział procesora do procesu odbywa się zgodnie z opisanym dalej algorytmem szeregowania. Procesy nieaktywne można podzielić na te, które czekają na zaistnienie pewnego zdarzenia, czyli na sygnał od innego procesu (tzw. procesy zawieszone) i te, które czekają na przydział procesora. Te ostatnie nazywa się procesami gotowymi.
Procesy gotowe tworzą kolejkę procesów gotowych, zlokalizowaną wewnątrz opisywanego modułu. Do kolejki tej nie ma bezpośredniego dostępu z zewnątrz. W chwili zwolnienia procesora przez aktualnie wykonywany proces, procesor jest przekazywany do pierwszego gotowego procesu lub zwracany procesowi wstrzymanemu przerwaniem. Procesy wstawiane są do kolejki procesów gotowych zgodnie z ich priorytetami, tzn. proces wstawiany jest za wszystkimi, które mają ten sam lub wyższy priorytet.
Warto podkreślić, że z zasady procesor zwalniany jest dobrowolnie. Nie ma możliwości odebrania go procesowi, nie używając przerwań. W przypadku przerwań przyjęto założenie, że proces inicjowany przerwaniem, podczas zwalniania procesora przekazuje go procesowi przerwanemu, a nie pierwszemu gotowemu.

Operacja **NEWPRO** z modułu SYSTEM zastąpiona jest w module MANAGER przez:

```
PROCEDURE STARTPROCESS (P: PROC; N: INTEGER;
PRIORITY: INTEGER;
REGULAR: BOOLEAN);
```

gdzie: **PROC** - wskazuje procedurę stanowiącą treść nowo tworzonego procesu, **N** - zawiera obszar potrzebny na dane procesu,

```
MODULE KANAL4;

(*VERSION 2.2.*)
(*DEFINITION MODULE MANAGER*)
  TYPE
    PROC=WORD;
    SIGNAL=WORD;
(*END MANAGER*)

CONST
  MOUT=04; (* adres rejestru 'data' dla 8255 *)
  MSTT=07; (* adres rejestru 'control' dla 8255 *)
  M=20;
  POZIOM=4;
  COMMAND=173;
  STINTA=13;
  CR=13; LF=10;

(*$E-*)

VAR
  PUSTY, PELNY: SIGNAL;
  POCZATEK, KONIEC, LZ: INTEGER;
  BUFOR: ARRAY[1..M] OF CHAR;

EXTERNAL PROCEDURE DISABL;
EXTERNAL PROCEDURE ENABLE;

(*DEFINITION MODULE MANAGER*)
  EXTERNAL PROCEDURE SEND (VAR S:SIGNAL);
  EXTERNAL PROCEDURE WAIT (VAR S:SIGNAL);
  EXTERNAL PROCEDURE DOIO (V: BYTE);
  EXTERNAL FUNCTION AWAITED (S:SIGNAL): BOOLEAN;
  EXTERNAL PROCEDURE STARTPROCESS (P:         PROC;
                                   N:         INTEGER;
                                   PRIORITY:  INTEGER;
                                   REGULAR:   BOOLEAN);
  EXTERNAL PROCEDURE INITMAN;
(*END MANAGER*)

PROCEDURE KONSUMENT;  * Procedura inicjowana jako proces *
VAR
  CH: CHAR;
BEGIN
  WHILE TRUE DO BEGIN
    IF LZ=0 THEN WAIT(PELNY);
    CH:=BUFOR[POCZATEK];
    IF POCZATEK=M THEN POCZATEK:=1 ELSE POCZATEK:=POCZATEK+1;
    LZ:=LZ-1;
    SEND(PUSTY);
    DOIO(POZIOM);
    OUT(MOUT):=CH;
  END
END (*KONSUMENT*

  *$E+*)

PROCEDURE STRWR4 (DL: INTEGER; S: STRING);
VAR
  I: INTEGER;
BEGIN
  FOR I:=DL-LENGTH(S) DOWNTO 1 DO WSTAW4(' ');
  FOR I:=1 TO LENGTH(S) DO WSTAW4(S[I]);
END (*STRWR4*);

PROCEDURE CRLFWR4;
BEGIN
  WSTAW4(CHR(CR)); WSTAW4(CHR(LF));
END (*CRLFWR*);

PROCEDURE WSTAW4 (CH: CHAR);
BEGIN
  IF LZ=M THEN WAIT(PUSTY);
  DISABL;
  BUFOR[KONIEC]:=CH;
(* BUFOR, KONIEC, LZ są danymi współdzielonymi i musi być
zapewniony wzajemnie wyłączny dostęp do nich *)
  IF KONIEC=M THEN KONIEC:=1 ELSE KONIEC:=KONIEC+1;
  LZ:=LZ+1;
  SEND(PELNY);
END (*WSTAW3*);

PROCEDURE INIKAN4;
BEGIN
  OUT(MSTT):=COMMAND;
  OUT(MSTT):=STINTA;
  POCZATEK:=1; KONIEC:=1; LZ:=0;
  PUSTY:=NIL; PELNY:=NIL;
  STARTPROCESS(WRD(ADDR(KONSUMENT)),36, 10, FALSE);
END (*INIKAN3*);

MODEND.
```

PRIORITY - określa względną ważność procesu wśród innych procesów,
REGULAR - wskazuje czy proces może współpracować z urządzeniami zewnętrznymi i wykorzystywać przerwania.
Nowo tworzony proces jest uaktywniany, natomiast proces realizujący tę operację jest wstrzymywany i wstawiany do kolejki procesów gotowych. Kolejka ta jest zadeklarowana jako lokalna w module MANAGER i niedostępna bezpośrednio na zewnątrz tego modułu. Umieszczone są w niej, w kolejności malejących priorytetów wszystkie procesy czekające na przydział procesora, tzn. gotowe do realizacji.

Jeśli dalsza realizacja procesu jest zależna od spełnienia pewnego warunku, to proces powinien zostać wstrzymany, zwalniając tym samym procesor i czekać na sygnał związany z tym warunkiem. Realizuje się tę operację za pośrednictwem:
PROCEDURE WAIT (VAR S: SIGNAL);
Proces, po jej wykonaniu, jest wstrzymywany i wstawiany do kolejki procesów wskazywanej przez S. Procesy wstawiane są z uwzględnieniem ich priorytetów, tzn. proces wstawiany jest do kolejki za wszystkimi procesami o priorytecie wyższym i równym.
Spełnienie warunku związanego z pewnym sygnałem S jest realizowane za pośrednictwem:
PROCEDURE SEND (VAR S:SIGNAL);
Jeśli sygnał został wysłany z procesu urządzenia, to pierwszy proces znajdujący się w kolejce **S** jest z niej usuwany i wstawiany do kolejki procesów gotowych, czekających na zwolnienie procesora. Wyróżnienie procesów urządzeń w tej operacji jest związane z koniecznością wznowienia realizacji procesu przerwanego przez proces urządzenia. Procesy urządzeń powinny mieć najwyższy priorytet, ponieważ obsługują urządzenia zewnętrzne, a brak tej obsługi jest równoważny z reguły ze wstrzymaniem jego dalszej pracy. Dlatego też proces, który realizuje wysłanie sygnału jest kontynuowany, gdy jest procesem urządzenia. W pozostałych przypadkach proces jest wstrzymywany i wstawiany do kolejki procesów gotowych.
Wykorzystanie procedury **IOTRAN** do wskazywania miejsca, w którym proces urządzenia zwalnia procesor na czas realizacji zainicjowanej w urządzeniu zewnętrznym operacji, jest bardzo niewygodne i może prowadzić do bardzo trudnych do wykrycia błędów. W miejsce tej operacji proponuje się:
PROCEDURE DOIO (V: BYTE);
Jest to logiczny równoważnik operacji **WAIT**. Powoduje ona zawieszenie aktualnego procesu do czasu wystąpienia wskazanego parametrem V przerwania sprzętowego, sygnalizującego wystąpienie określonej sytuacji w urządzeniu zewnętrznym.
Ponieważ typ **SIGNAL** jest ukryty na zewnątrz modułu MANAGER, wszystkie operacje na danych tego typu powinny być realizowane za pośrednictwem operacji zdefiniowanych wewnątrz modułu. Kolejną proponowaną operacją pozwalającą na sprawdzenie stanu kolejki procesów czekających na sygnał S jest
FUNCTION AWAITED (S:SIGNAL): BOOLEAN;
przyjmuje ona wartość **TRUE** jeśli kolejka nie jest pusta.

Przykład zastosowania

W przykładzie (wydruk) pokazane zostało rozwiązanie typowego zagadnienia współpracy dwóch procesów, z których jeden - nazywany dalej producentem - tworzy ciągi znaków (teksty), a drugi - proces konsumenta - odbiera je. W tym konkretnym przypadku konsumentem jest proces obsługi drukarki.
Postawiony problem można elegancko i efektywnie rozwiązać wykorzystując operacje zawarte w module MENAGER. W przykładzie założono, że proces producenta został umieszczony w innym module i wykorzystuje procedury modułu KANAL4.

Podsumowanie

Proponowany moduł MANAGER, zaprojektowany na bazie modułu SYSTEM, daje zestaw wygodnych w użyciu operacji pozwalających strukturalnie zapisać, w kategoriach języka wysokiego poziomu, niemal dowolny algorytm współbieżny. Wybrane elementy języka Pascal/MT+, w powiązaniu z proponowanym rozwiązaniem, dają możliwość tworzenia programów pracujących w czasie rzeczywistym i sterujących pracą różnych urządzeń zewnętrznych. Stosując przedstawione rozwiązanie zaprojektowano i wdrożono kilka systemów mikrokomputerowych czasu rzeczywistego. Pierwszym był układ do automatycznego pomiaru przekładników na taśmie produkcyjnej. Algorytm programu został zapisany w języku Pascal, a następnie przeniesiony na Assembler. Uzyskano w ten sposób zapis na tyle czytelny, że dalsze prace nad modernizacją poszczególnych funkcji mogły przebiegać bez większych trudności i bez udziału autora programu. System od kilku lat pracuje w przemyśle. Kolejny system, w którym w pełni zastosowano zaproponowaną koncepcję, to mikrokomputerowy układ zbierania i przetwarzania danych w kopalni węgla brunatnego. Czas realizacji projektu tego systemu był niespodziewanie krótki (około 4 miesięcy). System ten jest eksploatowany od ponad roku. Opisana koncepcja została także wykorzystana do konstrukcji oprogramowania dla systemu sterującego procesami produkcji sody kaustycznej w przemyśle papierniczym.
Dobre wyniki uzyskane w przypadku dotychczasowych wdrożeń pozwalają przypuszczać, że omawiane rozwiązanie może być szeroko stosowane. Obecnie trwają prace nad stworzeniem podobnych rozwiązań dla sterowników modułowych DIALOG z wykorzystaniem języka MODULA-2.

Mariusz Postół

EFC KOMPLEX

PRZEDSIĘBIORSTWO ZAGRANICZNE
KOMPLEX EFC

OFERUJE:

- **profesjonalne systemy mikrokomputerowe COMPLEX - XT/AT/386 konfigurowane zgodnie z życzeniem zamawiającego**
- **jedyny profesjonalny wielodostępny system operacyjny COMUNIX, umożliwiający pracę wieloterminalową i sieciową.**

System jest wyposażony w pełen zestaw oprogramowania narzędziowego z wielodostępną bazą danych włącznie.

Zapraszamy do korzystania z naszych usług w zakresie dostaw sprzętu, oprogramowania, doradztwa i szkoleń.

Zakład Elekroniki
61-706 Poznań
ul. Libelta 6
tel. 221-793

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Zbigniew Guzik

Forth – niedoceniony

Forth jest językiem wokół którego powstało wiele kontrowersji. Można by sądzić, że z chwilą gdy pojemności pamięci komputerów osobistych są już określane w megabajtach - przejdzie do historii. Tymczasem jego popularność wzrasta, lecz nie wśród użytkowników komputerów osobistych, a wśród projektantów systemów sterowania (mi .).

Forth jest zbiorem różnych koncepcji programowych, tworzących spoistą całość. Jest on:
- językiem wyższego poziomu, jednocześnie zawiera rozkazy z poziomu języka maszynowego,
- językiem interpretacyjnym, a zarazem kompilatorem,
- wielozadaniowym systemem operacyjnym,
- językiem samorozszerzalnym, pozwalającym definiować nowe procedury i biblioteki aplikacyjne na podstawie już istniejących i wykorzystywać je jak instrukcje,
- edytorem i zestawem środków uruchomieniowych,
- językiem meta-aplikacyjnym, umożliwiającym tworzenie nowych, problemowo zorientowanych języków.

Forth opisany jest za pomocą Forth'a. Może się to wydać dziwne, ale dla objaśnienia języka polskiego używamy polskich słów. Zdefiniowanie Forth'a w ten sposób upraszcza przenoszenie systemu między różnymi komputerami. Cały język (poza kilkunastoma bajtami tworzącymi środowisko wirtualne) to słownik, składający się z niezależnych, połączonych między sobą w hierachiczną, drzewiastą strukturę procedur, nazywanych słowami. Każdemu słowu odpowiada nazwa, przypisany jej program wykonawczy i parametry. Z wyjątkiem kilkudziesięciu słów tworzących jądro języka, wszystkie kolejne są tworzone z użyciem uprzednio zdefiniowanych, prostszych słów. Programowanie polega na definiowaniu kolejnych słów do momentu, kiedy pojedyncze słowo finalne opisuje całe zadanie.

Przypuśćmy, że trzeba zamienić miejscami dwie liczby na szczycie stosu. Najpierw należy zdefiniować operacje **POPA**, **PUSHA**, **POPB**, **PUSHB** przepisujące wartości między wierzchołkiem stosu a rejestrami A i B. Żądanie zamiany można zrealizować wprowadzając na terminal sekwencję:

POPA POPB PUSHA PUSHB ENTER

Można też zdefiniować nowe słowo **SWAP**:

: SWAP POPA POPB PUSHA PUSHB; ENTER

Słowa **:** i **;** oznaczają początek i koniec operacji definiowania nowego słowa. Zamiana dwóch liczb na stosie może być teraz zrealizowana pojedynczym słowem **SWAP**. Każda nowa definicja jest kompilowana w trakcie wprowadzania i jest natychmiast gotowa do wykonania.

W języku Forth słowa zasadniczo komunikują się między sobą przez stos, sterowanie którym jest jawne i bezpośrednie (można ze stosu pobierać parametry, jak i umieszczać na stosie rezultaty). Konsekwencją wykorzystania stosu jest przyjęcie w zapisie wyrażeń Odwrotnej Polskiej Notacji (znanej m. in. użytkownikom kalkulatorów Hewlett-Packard). Argumenty muszą tu pojawiać się przed odpowiednimi operatorami. Np. wyrażenie (1+2)*(3+4) w języku Forth zapisywane jest:
1 2 + 3 4 + *.

Implementacja

Forth redukuje objętość podprogramów do minimum, gdyż cały język jest zbudowany z procedur, mających charakter wywołań podprogramów:

```
call E ----> E:     A
call F              B
call G              C
                    ret
```

Jeżeli cały program jest serią wywołań, to można wyeliminować kod rozkazu **call** stosując niewielki program (interpreter adresów); **ip** jest tu wskaźnikiem adresów, a **()** oznacza odwołanie pośrednie

```
NEXT1:  ip := ip+1
        call (ip)
        jmp NEXT1
```

Można też wyeliminować stos powrotów dla sekwencji **call..ret** przez kończenie każdego programu kodem **jmp NEXT2** i następującą modyfikacją interpretera adresów.

```
F:  xxx          NEXT2   ip := ip+1
    jmp NEXT2            jmp (ip)
```

Wywoływany program może być listą adresową kolejnych podprogramów. W tym wypadku pierwszemu elementowi takiej listy przypisuje się rolę wskaźnika specjalnego programu, nazywanego interpreterem wewnętrznym **NEST**. Zawiesza on interpretację listy głównej, zapisując bieżący adres na stos powrotu. W przypadku, gdy wywoływany program jest kodem maszynowym, wskaźnik interpretera zawiera adres początku kodu. Ostatnim elementem wywoływanej listy powinien być wskaźnik programu **UNNEST**, zapewniającego powrót do listy głównej. Opisana technika nosi nazwę pośredniego kodu nizanego (ang. indirect thereaded code). Ostateczne postaci interpreterów są następujące:

```
NEXT:     ip := ip+1
          w := (ip)
          jmp (w)
NEST:     sp := sp-1
          (sp) := ip
          jmp NEXT
UNNEST:   ip := (sp)
          sp := sp+1
          jmp NEXT
```

Wskaźnik interpretera wewnętrznego (pole kodu) słowa może zawierać adres interpretera traktującego pozostałą część listy (pole parametrów) w inny sposób. Pole kodu, pole parametrów, identyfikator (zawierający nazwę słowa) i łącznik (umożliwiający przeszukiwanie słownika), tworzą słowo języka Forth. Generalnie, pole kodu - tworzone przez słowo definiujące - wskazuje na program, który decyduje, w jaki sposób wykonane będzie słowo i jak wykorzystane będą jego parametry. Przykładami interpreterów wewnętrznych są **DOCON** i **DOVAR**. Interpreter **DOCON** traktuje pole parametrów jako listę stałych i umieszcza ją na stosie. **DOVAR** umieszcza na stosie adres pola parametrów, wykorzystywanego jako miejsce przechowywania zmiennej. Dla umożliwienia dostępu do pola parametrów wprowadzony jest rejestr „w". Należy zaznaczyć, że Forth wykorzystuje dwa stosy - stos danych i stos powrotu, obsługujący listę adresową.

Jak działa Forth?

Zbiór słów (słownik), algorytmy postępowania, urządzenia zewnętrzne (terminal, dysk, drukarka), procesor i pamięć operacyjna tworzą środowisko, w którym wykonywany jest program. Jest on ciągiem znaków ASCII i nie znajduje się w pamięci operacyjnej, lecz wprowadzany jest z terminala lub z dysku. Instrukcje wewnętrzne słowa są odpowiednikiem mikrokodu rozkazu maszynowego. Jedynym znakiem sterującym w języku Forth jest spacja, fizycznie oddzielająca jedno słowo od drugiego. Inne znaki sterujące, (np. powrót karetki), są interpretowane tylko na poziomie wejścia-wyjścia.

System zawsze znajduje się w jednym z dwóch stanów - wykonawczym lub kompilacyjnym. W stanie wykonawczym, Forth po wprowadzeniu kolejnego słowa, przeszukuje słownik i jeśli znajdzie identyczne, wykonuje je. Jeśli nie - próbuje zinterpretować zapis jako liczbę i umieścić ją na stosie. Jeśli i to nie jest możliwe - generowany jest komunikat o błędzie. W trybie kompilacji, wprowadzane kolejno definicje słów są umieszczane w słowniku. Przejście do trybu kompilacji następuje najczęściej po użyciu słowa definiujące np. **CONSTANT**, **VARIABLE** lub :(dwukropek).

Opisana sekwencja, definiująca zamianę dwóch wartości ze stosu, umieszcza w słowniku nowe słowo **SWAP**, pole kodu adresujące interpreter **NEST** oraz pole parametrów z listą adresową czterech słów **PUSH/POP** i adres programu **UNNEST**. Interpreter zewnętrzny działa w nieskończonej pętli wprowadzającej tekst z terminalu lub z pliku dyskowego.

Kompletny system Forth zajmuje około 8 KB pamięci operacyjnej (Pascal potrzebuje ok. 48 KB). Jeśli konieczna jest oszczędność pamięci, np. program sterujący procesem przemysłowym musi być zapisany w pamięć PROM, to ok. 90% języka (nazwy słów, interpreter tekstu, edytor i in.) może zostać wyeliminowane.

Forth jest rekursywny i wielobieżny (ang. reentrant), tzn. dowolne słowo może wywoływać samo siebie, te same procedury mogą być jednocześnie wykorzystywane przez różne segmenty programu (np. przez program główny i procedurę obsługi przerwania).

Szybkość przetwarzania w języku Forth jest zazwyczaj dość duża. Zależy ona od efektywności tworzonego w języku maszynowym programu **NEXT**. Ocenia się, że Forth działa tylko dwukrotnie wolniej od języka maszynowego, przy czym najbardziej krytyczne czasowo procedury mogą być zapisane bezpośrednio w języku maszynowym. Słowo **CODE** wywołuje Assembler i rozpoczyna kompilację na kod maszynowy.

W języku Forth można stosować przetwarzanie współbieżne. Najczęściej jest to realizowane słowem **PAUSE**, które przekazuje sterowanie następnemu w kolejce programowi.

Forth umożliwia też tworzenie konstrukcji strukturalnych:

IF ... ELSE ... THEN
BEGIN ... UNTIL
BEGIN ... WHILE ... REPEAT
DO ... LOOP lub DO ... +LOOP

Nie mogą one być wykonywane w trybie konwersacyjnym, lecz muszą być kompilowane w nowo zdefiniowanych słowach. Możliwe jest też zagnieżdżanie struktur. Forth nie dopuszcza instrukcji **GOTO** ani etykiet.

Arytmetyka

Wokół języka Forth narosło wiele kontrowesji. Jedną z nich są obliczenia zmiennoprzecinkowe. Forth nie ma sformalizowanej reprezentacji typów danych. Wszystkie parametry przygotowywane są na stosie w postaci wielokrotności 16-bitowego słowa maszynowego i zadaniem procedury (słowa) jest właściwa interpretacja tych danych (liczba całkowita ze znakiem lub bez znaku, liczba podwójnej precyzji, wartość logiczna, znak ASCII, adres komórki pamięci itd.). Słowo 16-bitowe jest najmniejszą stosową jednostką i np. wartość logiczna **TRUE** ma postać $FFFF (czyli -1), a znak ASCII jest uzupełniany zerowym, bardziej znaczącym bajtem. Bajtowe łańcuchy znakowe (string) mogą być przedstawiane w tzw. postaci liczonej (pierwszy znak łańcucha opisuje jego rozmiar), a na stosie reprezentowane są przez adres ich początku w pamięci.

Zazwyczaj stosowanie zmiennego przecinka jest niejako automatyczne - a programista przyjmuje, że troszczenie się o miejsce przecinka w liczbie jest zadaniem komputera. W języku Forth podstawowym kryterium jest szybkość. Różnice w szybkości wykonywania analogicznych operacji przy stałym i zmiennym przecinku (bez koprocesora arytmetycznego) są znaczne. Jednak nawet najbardziej złożone operacje mogą być wykonane z użyciem stałego przecinka, przy zastosowaniu techniki skalowania. Dlatego też w podstawowej wersji

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Forth'a zastosowano wyłącznie stały przecinek, dając w zamian użytkownikowi unikalny zestaw słów, zwanych operatorami skalującymi. Technikę skalowania można prześledzić na przykładzie stosowania słowa */, które mnoży dwie liczby ze stosu i dzieli rezultat przez trzecią, przy czym wartość pośrednia obliczana jest z podwójną precyzją. Przykładowo, większość komputerów oblicza 2/3 liczby 234 następująco: 0,66666*234=155,99999. Wynik, mimo stosowania zmiennego przecinka, nie jest dokładny i wymaga zaokrąglenia. Stosując Forth postępujemy następująco: 234*2=468; 468/3=156, co w zapisie języka ma postać: 234 2 3 */
Wynik wyliczony jest szybciej , prościej i dokładniej. Stosując filozofię języka Forth można przedstawić z dużą dokładnością dowolne stałe np. π=355/113, e=26667/10546.
Opisana metoda nie wyklucza stosowania zmiennego przecinka, który może być z łatwością zrealizowany w formie podprogramów napisanych w języku maszynowym.

Pamięć wirtualna

W języku Forth pamięć dyskowa jest integralną częścią środowiska. Użytkownik może adresować dane na dysku w podobny sposób, jak pamięć operacyjną, nie troszcząc się o sposób dostępu do nich. Jest to koncepcja tzw. pamięci wirtualnej. Pamięć dyskowa podzielona jest na kolejne, ponumerowane bloki o stałej objętości. W pamięci operacyjnej zarezerwowane są specjalne bufory, przeznaczone na czasowe przechowywanie zawartości bloków. Potrzebny blok odczytywany jest z dysku do jednego z buforów. Jeśli wszystkie bufory są już zapełnione, system wybiera najwcześniej zajęty bufor i udostępnia go, uprzednio przepisując jego zawartość na dysk (jeśli była zmodyfikowana). Słowem **FLUSH** mogą zostać zapisane na dysku wszystkie zmodyfikowane bloki. W ten sposób można zapewnić ochronę danych.
Jeżeli wiadomo w jaki sposób zapisać i odczytać sektor dyskowy, implementacja pamięci wirtualnej Forth'a nie nastręcza większych kłopotów. Forth nie potrzebuje więc zewnętrznego systemu operacyjnego, lecz tworzy swój własny. Współdziałanie z terminalem i drukarką organizuje się przez tworzenie oddzielnych słów napisanych w języku maszynowym.
Można też zastosować istniejący już system operacyjny (CP/M czy MS-DOS), wykorzystując jego wywołania BDOS. Obniża to jednak wydajność systemu.

Forth-83

Forth został opracowany w latach sześćdziesiątych przez C. Moore'a, podczas jego prac nad sterowaniem urządzeniami astronomicznymi w Narodowym Obserwatorium Radio-Astronomicznym w USA. Język rozpowszechnił się szybko i w 1974 r. został uznany za standard przez Międzynarodową Unię Astronautyczną. Wersja języka opracowana przez autora nazywała się poly-Forth. Entuzjaści języka utworzyli zespół FIG (Forth Interest Group), wydając czasopismo (Forth Bulletin) i udostępniając bezpłatnie wersje języka fig-Forth dla popularnych mikroprocesorów. Jednocześnie powstał zespół normalizacyjny (Forth Standard Team), rezultatem pracy którego było opublikowanie standardowego zestawu słów FORTH-79, a następnie FORTH-83 (patrz tabela). Obecnie wersje Forth'a istnieją dla wszystkich liczących się komputerów. Nieświadomy kontakt z językiem mają ci, którzy z upodobaniem zajmują się grami video, bowiem większość gier została przygotowana właśnie w języku Forth. Znakomitą wersją języka jest system F-83, opracowany przez H. Laxena i M. Perry.
Podstawowe udoskonalenia ostatniej wersji standardu to:
- monoadresowanie, czyli ustalenie jedynego adresu dla bezpośrednich odwołań do słów,
- wyeliminowanie słów, których działanie zależne jest od stanu interpretera tekstu (np. **LITERAL**, **."**, **'**, **(**) i rozdzielenie ich na stanowo niezależne,
- udoskonalenie działania pętli **DO-LOOP**,
- zasada zaokrąglania rezultatu dzielenia w kierunku ujemnej nieskończoności,
- zmiana reprezentacji wartości logicznej **TRUE** (obecnie -1).

Oprócz tego standard definiuje listę słow podwójnej precyzji, kontrolowanych, niekontrolowanych i zestaw zaleceń.

Tabela. Standardowe słowa języka FORTH-83

Operacje logiczne	NOT AND OR XOR
Manipulacje na stosie	DUP DROP SWAP OVER ROT PICK ROLL ?DUP >R R> R@ DEPTH
Relacje	< = > 0< 0= 0> D< U<
Operacje arytmetyczne	+ D+ - 1+ 1- 2+ 2- * UM* UM/MOD 2/ / MOD /MOD */ MAX MIN ABS NEGATE DNEGATE
Operacje na pamięci	@ ! C@ +! C!
Łańcuchy	CMOVE CMOVE> FILL COUNT -TRAILING
Konwersja numeryczna	BASE DECIMAL CONVERT <# # #S HOLD SIGN #>
Struktury programowe	DO LOOP +LOOP LEAVE I J IF ELSE THEN BEGIN UNTIL WHILE REPEAT EXIT EXECUTE
Obsługa konsoli operatorskiej	KEY EMIT . U. ." .(CR TYPE SPACE SPACES SPAN EXPECT
Obsługa pamięci masowej	BLOCK BUFFER UPDATE FLUSH SAVE-BUFFERS
Wywoływanie i kończenie programu	FORTH-83 QUIT ABORT ABORT"
Adresowanie słownika	HERE PAD TIB >BODY
Kompilator i interpreter	LOAD (, ALLOT DOES> STATE IMMEDIATE [COMPILE] COMPILE LITERAL [ ] WORD BLK >IN #TIB
Słowniki	FORTH DEFINITIONS ' ['] FIND FORGET
Definiowanie słów	: ; CREATE VARIABLE CONSTANT VOCABULARY

Niedoskonałości Forth'a

Podstawową wadą Forth'a jest mała przejrzystość programów źródłowych. Przeciwnicy twierdzą, że jest to język **write only.** Zrozumiałość tekstu źródłowego można poprawić przez stosowanie dogłębnych komentarzy i dzielenie programu na niewielkie moduły logiczne.
Inne wady języka to: konieczność bezpośredniego gospodarowania stosem przez programistę, każdorazowe określanie typów danych operacji, brak katalogów programów źródłowych i brak kompresji znaków w blokach redagowanych tekstów. Do wad języka należy także jego mała przydatność w realizacji złożonych obliczeń numerycznych. Uważa się również, że tego wadą języka jest brak arytmetyki zmiennoprzecinkowej i grafiki. Struktura Forth'a pozwala jednak bez trudu wprowadzić odpowiednie rozszerzenia, co ma miejsce w większości oferowanych komercjalnie wersji.

Zastosowania

Ze względu na wymienione wady, Forth nie stanie się nigdy tak popularny, jak np. Pascal. Istnieje jednak dziedzina, w której Forth może okazać się niezastąpiony, tzn. sterowanie cyfrowe. W działalności inżynierskiej, jak i obsługa eksperymentów naukowych, sterowanie procesami przemysłowymi czy sterowanie inteligentnymi urządzeniami, to sfera ekspansji Forth'a.

Zbigniew Guzik

CONSULT

Biuro Usług Consultingowych „CONSULT" Sp. z o.o.
Gdańsk 6, skrytka pocztowa 48
tel. 51-69-21, tlx. 512416 cons pl

Consulting i wykonanie ekspertyz w zakresie [illegible] informatyki
w przedsiębiorstwach
programowanie użytkowe i narzędziowe oraz specjalistyczne z uwzględnieniem
specyfiki tematu
niezwłoczne terminy dostaw sprzętu typu IBM PC ATARI w dowolnej konfiguracji, plotery,
digitizery, karty sieci
serwis gwarancyjny 12-miesięczny oraz pogwarancyjny serwis mikrokomputerów
firmy „ATARI"
usługi w zakresie wykonywania obliczeń przestrzennych rozkładów zanieczyszczeń
w powietrzu atmosferycznym

literaturę i dokumentację producenta do dostarczonego sprzętu i oprogramowania

CONSULT to wszystko zapewnia

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

W polskiej literaturze informatycznej rzadko spotyka się opisy algorytmów arytmetyki zmiennoprzecinkowej. Jest to efekt uboczny rozwoju techniki komputerowej widzianej z perspektywy kopii IBM PC XT/AT - tu wykorzystywane jest gotowe oprogramowanie. Komputeryzacja to jednak nie tylko biura - to głównie przemysł, w którym urządzenia wytwórcze nie powinny być sterowane przez komputery osobiste, lecz odpowiednio przystosowane do trudnych warunków pracy systemy mikroprocesorowe.

Sinus, cosinus tangens

Jeżeli w mikrokomputerze nie jest stosowany koprocesor arytmetyczny, to operacje arytmetyczne muszą być realizowane z użyciem odpowiednich procedur zapisanych w języku maszynowym. Dla czterech podstawowych operacji arytmetycznych zazwyczaj przewidziane są oddzielne procedury, nie odwołujące się do innych podprogramów. Również do obliczania pierwiastka kwadratowego z reguły stosuje się specjalny, efektywny algorytm, który realizowany jest jako samodzielna procedura. W przypadku pozostałych funkcji arytmetycznych takich, jak logarytmy, funkcje wykładnicze czy trygonometryczne używa się algorytmów, które zakładają istnienie podprogramów realizujących pięć podstawowych operacji.
W przedstawionym dalej algorytmie obliczania funkcji trygonometrycznych, za podstawę optymalizacji przyjęto typowe (średnie) czasy wykonywania wymienionych pięciu podstawowych operacji (tabela). Warto zwrócić uwagę, że czas wykonywania mnożenia (t_3) ma zazwyczaj dominujący wpływ na czas realizacji całego algorytmu i z tego względu bardzo istotna jest jego optymalizacja.
Jeżeli przez $\mathbf{t_i}$ oznaczony zostanie czas wykonania i-tej z wymienionych operacji to czas $\mathbf{t}$, potrzebny na wykonanie całego algorytmu, można wyrazić wzorem:

$$t = t_0 + \sum_i n_i * t_i$$

gdzie $\mathbf{n_i}$ jest liczbą naturalną, oznaczającą krotność wystąpienia i-tej operacji, a t_0 odpowiada dodatkowym operacjom potrzebnym do realizacji algorytmu (z reguły $\mathbf{t_0}$ jest dość małe).
Przed przystąpieniem do omawiania poszczególnych algorytmów należy ustalić jeszcze format zapisu liczb. Późniejsze rozważania są w zasadzie niezależne od przyjętego formatu, ale wnioski wypływające z analizy czasowej będą różne dla poszczególnych formatów. Założono, że liczby są zapisywane w czterech bajtach (zapis zmiennoprzecinkowy, pojedyncza precyzja) tak, jak to przedstawiono na rysunku 1. Pierwszy bajt reprezentuje cechę **e** (exponent), zapisaną z nadmiarem +129 (+81H). W pozostałych trzech bajtach zapisywana jest 24-bitowa mantysa **m** (mantissa), znormalizowana do zakresu [1,2). Najbardziej znaczący bit mantysy jest zawsze równy 1, dzięki czemu można go zastąpić znakiem ■ (sign) liczby. Przyjęto, że s=0 dla „+" oraz s=1 dla „−". Wówczas:
liczba=(-1)^s*m*2^(e-129)
W podanym formacie nie można zapisać zera, dlatego przyjmuje się, że zero jest reprezentowane przez **e**=0 i wówczas bity mantysy są pomijane.
Proponowany zapis ogranicza zbiór liczb rzeczywistych do pewnego dyskretnego i skończonego podzbioru liczb wymiernych (z zakresu około ± 2,9E-39 do 1,7E+38) i zapewnia dokładność do siedmiu cyfr znaczących.

Dla przykładu, -5.75 zostanie zapisane jako :
83 B8 00 00 (szesnastkowo)
natomiast π=3.1415926..
82 49 0F DB

W omówionych dalej algorytmach obliczania wartości funkcji sinus, cosinus i tangens założono, że argument został już sprowadzony do zakresu $[0, \pi/4]$. W artykule pominięto problem przechodzenia z dowolnymi argumentami do podanego zakresu.
Analizując problem numerycznego wyznaczania wartości funkcji w zadanym punkcie należy rozważyć trzy metody. Do najprostszych sposobów można zaliczyć obliczenie wartości dla rozwinięcia funkcji w szereg Taylora. Szereg Taylora dla podanego zakresu argumentów jest jednak zbyt wolno zbieżny, aby dać zadowalające wyniki. Z tego względu dość powszechnie stosuje się różne metody znajdowania optymalnych wielomianów aproksymujących zadaną funkcję, przy założeniu określonego zakresu argumentów i wymaganej dokładności wyniku. Najbardziej rozpowszechniona jest metoda tzw wielomianów Czebyszewa. Zazwyczaj wystarcza jedynie znajomość odpowiedniego wielomianu dla tangensa, z którego można potem wyznaczyć sinus i cosinus. Uwzględniając parametry czasowe podane w tabeli i wykorzystując różne możliwe wielomiany, otrzymuje się:

$$\sum_i n_i * t_i = 3ms$$

Czasami stosowana jest jeszcze inna metoda sprowadzająca się do wykorzystaniu warteści sinusa i cosinusa dla pewnych argumentów, co pozwala na skorzystanie ze wzoru:

sin(x+y+z)=sin(x)*cos(y)*cos(z)-sin(y)*sin(z)+cos(x)*(sin(y)*cos(z)+cos(y)*sin(z))

gdzie x,y,z reprezentują poszczególne bajty mantysy. Nie wchodząc w szczegóły należy stwierdzić, że rozważana metoda nie pozwala na osiągnięcie lepszego wyniku, niż:

$$\sum_i n_i * t_i = 2,5ms$$

i wymaga użycia tablicy zajmującej znaczny obszar pamięci (rzędu (3+3)*256*4 = 6 KB. Zaletą tej metody jest możliwość stosowania jej w każdym przypadku, bez konieczności wyznaczania odpowiednich wielomianów, a jedynie w oparciu o znane wartości funkcji w pewnych punktach. Proponowany przez autora sposób jest kombinacją metod wykorzystujących tablice i szereg Taylora.

Dowolny argument $x \in [0, \pi/4]$ można zapisać:
x=y+z
gdzie $y \in \{0, 1/64, 2/64, \ldots, 50/64\}$
oraz $z \in [0, 1/64)$
Zauważmy, że $51/64 > \pi/4$
Wiedząc, że :

sin(x)=sin(y+z)=sin(y)*cos(z)+cos(y)*sin(z)

można obliczyć wartość podanego wyrażenia biorąc sin(y) i cos(y) z odpowiednich tablic, oraz wyliczając sin(z) i cos(z) za pomocą szeregu Taylora. Procedura realizująca ten algorytm będzie wymagała przechowywania na stałe dwóch tablic, tj.:
sin(0), sin(1/64), sin(2/64), ... , sin(50/64)
cos(0), cos(1/64), cos(2/64), ... , cos(50/64)
Łącznie daje to 408 bajtów. Pozostaje pytanie, jak wiele wyrazów szeregu Taylora należy uwzględnić, aby osiągnąć dla $z \in [0, 1/64]$ wymaganą dla pojedynczej precyzji dokładność wyniku (błąd względny rzędu 2^{-24}). Jeżeli zostanie przyjęte (początkowe dwa wyrazy szeregu Taylora):

$$\sin(z) = z - \frac{1}{3!} * z^3$$

$$\cos(z) = 1 - \frac{1}{2!} * z^2$$

to uzyskamy dla $z \in [0,1/64]$:

$$\frac{\frac{1}{5!} * z^5}{z - \frac{1}{3!} * z^3} < z^4$$

$$\frac{\frac{1}{4!} * z^4}{1 - \frac{1}{2!} * z^2} < z^4$$

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Powyżej użyto trzeciego wyrazu szeregu Taylora, jako promienia zbieżności, co jest kryterium znacznie silniejszym od standardowo przyjmowanej tzw. postaci Lagrange'a reszty R_n, określającej promień zbieżności szeregu. Jednakże w rozwinięciu funkcji sinus i cosinus w szereg Taylora, każdy następny wyraz jest, co do wartości bezwzględnej, mniejszy od poprzedniego, a jego znak jest przeciwny, co daje:

$$|\sin(z) - (z - \frac{z^3}{3!})| \le \frac{1}{5!} * |z^5|$$

$$|\cos(z) - (1 - \frac{z^2}{2!})| \le \frac{1}{4!} * |z^4|$$

Ostatecznie uzyskuje się następujące wzory:

$$\sin(x) = \sin(y) * (1 - \frac{z^2}{2!}) + \cos(y) * (z - \frac{z^3}{3!})$$

$$\cos(x) = \cos(y) * (1 - \frac{z^2}{2!}) - \sin(y) * (z - \frac{z^3}{3!})$$

Czas realizacji przedstawionego algorytmu można oszacować przy dodatkowych założeniach, że dzielenie przez 3 można zastąpić mnożeniem przez 1/3, natomiast dzielenie przez 2 jest równoznaczne zmniejszeniu o 1 cechy ■, a tym samym czas wykonywania tej operacji można pominąć. W efekcie daje to 5 mnożeń, 2 odejmowania i 1 dodawanie, czyli:

$$\sum_i n_i * t_i = 2,2ms$$

Przedstawiony algorytm zastosowano dla mikroprocesora Z80, który po uwzględnieniu dodatkowych operacji, pozwolił uzyskać łączny czas rzędu 2,7 ms.

Przy obliczaniu tangensa korzysta się ze wzoru:

$$\tan(x) = \frac{\sin(x)}{\cos(x)}$$

oraz z faktu, że podczas obliczania cos(x) można skorzystać ze znanych już wartości sin(z) i cos(z), wyznaczonych przy liczeniu sin(x). Mając do wykonania 7 mnożeń, 3 odejmowania, 1 dodawanie i 1 dzielenie otrzymuje się:

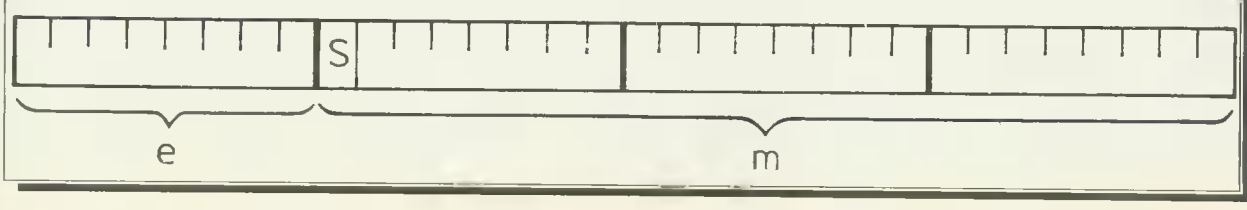
$$\sum_i n_i * t_i = 3,6ms$$

oraz łączny czas wykonywania całego algorytmu rzędu 4,7 ms.

Wojciech Skaba

Tabela: Typowe czasy wykonywania operacji arytmetycznych podane w milisekundach (Z80 - 4 MHz)

Procedura	Symbol	Czas [ms]
dodawanie	t_1	0,10
odejmowanie	t_2	0,16
mnożenie	t_3	0,35
dzielenie	t_4	0,56
pierwiastek	t_5	1,18



Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Jan Bielecki

TURBO-C

Programowanie operacji we-wy

Operacje we-wy dotyczą zbiorów danych albo urządzeń. W programach zarówno zbiory danych jak i urządzenia są reprezentowane przez pliki. Plik nie jest obiektem fizycznym, lecz stanowi tylko pewien aspekt, logiczną interpretację obiektu fizycznego jakim jest zbiór danych w pamięci masowej albo zbiór danych z urządzenia znakowego (takiego jak np. klawiatura).
Koncentrując się na zakodowaniu algorytmu w języku wysokiego poziomu, wygodnie jest posługiwać się abstrakcyjnym pojęciem pliku. Należy jednak pamiętać, że zawarte w algorytmie operacje na plikach będą podczas wykonywania programu realizowane jako operacje na zbiorach i urządzeniach. Z tego względu, wykonywanie operacji na pliku musi być poprzedzone ustanowieniem pliku, a następnie skojarzeniem go z ustalonym zbiorem danych.
W Turbo C do ustanowienia i skojarzenia pliku ze zbiorem służy funkcja fopen, a do usunięcia takiego skojarzenia - funkcja fclose. Rezultatem funkcji fopen jest wskazanie danej typu FILE - zdefiniowanego w zbiorze stdio.h . Argumentem funkcji fclose jest natomiast wskazanie udostępnione przez funkcję fopen. Każde z wymienionych wskazań jest nazywane wskazaniem plikowym i umożliwia identyfikację pliku.
Poza wskazaniami udostępnianymi za pomocą funkcji fopen, w zbiorze stdio.h zdefiniowano dodatkowe wskazania plikowe, reprezentowane przez makrodefinicje: stdin, stdout i stderr. Służą one odpowiednio do identyfikowania standardowego pliku wejściowego, standardowego wyjściowego oraz pliku do sygnalizowania błędów.

Funkcja fopen

Nagłówek: FILE *fopen(**name**,**mode**)
char *__name__,*__mode__;

Wykonanie funkcji fopen powoduje ustanowienie pliku, skojarzenie go ze zbiorem o nazwie określonej przez **name** i udostępnienie unikalnego wskazania plikowego identyfikującego plik. Jeśli **mode** reprezentuje ciąg znaków **r**, to plik zostanie otwarty w trybie do wprowadzania, a jeśli **w**, to do wyprowadzania. Jeśli zbiór o podanej nazwie już istnieje, to tuż przed otwarciem pliku w trybie do wyprowadzania zostanie on usunięty. Gdy otwarcie pliku okaże się niemożliwe (np. plik jest otwierany do wprowadzania, a zbiór o podanej nazwie nie istnieje), to rezultatem funkcji jest wskazanie puste.

Funkcja fclose

Nagłówek: int fclose(**file**)
FILE *__file__;

Wykonanie funkcji fclose powoduje zamknięcie pliku identyfikowanego przez **file**. Rezultatem funkcji jest dana o wartości 0 - jeśli zamknięcie pliku było pomyślne albo - w przeciwnym razie dana o wartości EOF.
Podczas wykonywania operacji we-wy na plikach identyfikowanych przez wskazania plikowe, mogą wystąpić dwa rodzaje sytuacji wyjątkowych: błąd operacji oraz próba wprowadzenia danych z pliku znajdującego się w pozycji końcowej. Stosownie do przypadku, w strukturze typu FILE wskazywanej przez wskazanie plikowe zostaje odnotowana informacja, do której można odwoływać się za pomocą funkcji ferror i feof. Informacja ta może zostać usunięta za pomocą funkcji clearerr.

Funkcja ferror

Nagłówek: int ferror(**file**)
FILE *__file__;

Wykonanie funkcji ferror powoduje udostępnienie informacji o tym, czy podczas wykonywania operacji na pliku identyfikowanym przez **file** wystąpił błąd. Rezultatem funkcji jest wtedy dana o wartości różnej od 0.

Funkcja feof

Nagłówek: int feof(**file**)
FILE *__file__;

Wykonanie funkcji feof powoduje udostępnienie informacji o tym czy podczas wykonywania operacji na pliku identyfikowanym przez **file** podjęto próbę wprowadzenia danych wtedy, gdy plik znajdował się w pozycji końcowej. Rezultatem funkcji jest wtedy dana o wartości różnej od 0.

Funkcja clearerr

Nagłówek: void clearerr(**file**)
FILE *__file__;

Wykonanie funkcji clearerr powoduje usunięcie ze struktury wskazywanej przez **file**, informacji o ewentualnym błędzie operacji oraz próbie wprowadzenia danych z pliku, gdy znajdował się on w pozycji końcowej.

Funkcja fgetc

Nagłówek: int fgetc(**file**)
FILE *__file__;

Wykonanie funkcji fgetc powoduje wprowadzenie kolejnego znaku z pliku identyfikowanego przez **file**. Rezultatem funkcji jest dana o wartości równej kodowi wprowadzonego znaku. Jeśli w chwili wywołania funkcji fgetc plik znajduje się w pozycji końcowej, to rezultatem funkcji jest dana o wartości EOF.

Funkcja fputc

Nagłówek: int fputc(**c**,**file**)
int __c__;
FILE *__file__;

Wykonanie funkcji fputc powoduje wyprowadzenie do pliku identyfikowanego przez **file** znaku o kodzie **c**. Rezultatem funkcji jest dana, której wartością jest kod wyprowadzonego znaku. Jeśli wyprowadzenie znaku nie udało się, to rezultatem funkcji jest dana o wartości EOF. Wywołanie fputc(**c**,stdout) może być zastąpione wywołaniem putchar(**c**).
Przykład
Sposób użycia omówionych funkcji przedstawiono na wydruku 1. Wykonanie programu powoduje przeniesienie zawartości zbioru JANB.DOC na ekran monitora.
W programach znacznie częściej niż funkcje fgetc i fputc wykorzystywane są funkcje fread, fwrite, fprintf i fscanf. Zostaną one przedstawione w znacznym uproszczeniu, ponieważ chodzi jedynie o zilustrowanie zasad ich wykonywania.

Funkcja fread

Nagłówek: int fread(**buf,size,count,file**)
void *__buf__;
int __size__,
int __count__;
FILE *__file__;

Wykonanie funkcji fread powoduje wprowadzenie z pliku identyfikowanego przez **file**, do obszaru pamięci operacyjnej wskazywanego przez **buf, count** obiektów o rozmiarze **size** (każdy). Rezultatem funkcji jest dana o wartości równej liczbie wprowadzonych obiektów. Jeśli jest ona mniejsza od **count**, to wystąpił błąd wykonania funkcji. albo plik przedwcześnie znalazł się w pozycji końcowej.

Funkcja fwrite

Nagłówek: int fwrite(**buf,size,count,file**)
void *__buf__;
int __size__;
int __count__;
FILE *__file__;

Wykonanie funkcji fwrite powoduje wyprowadzenie do pliku identyfikowanego przez **file**, z obszaru pamięci operacyjnej wskazywanego przez **buf, count** obiektów o rozmiarze **size** (każdy). Rezultatem funkcji jest dana o wartości równej liczbie wyprowadzonych obiektów. Jeśli jest ona mniejsza od **count**, to jest to objawem błędu wykonania funkcji.

Funkcja fprintf

Nagłówek: int fprintf(**file,format, ...**)
FILE *__file__;
char *__format__;

Wykonanie funkcji fprintf powoduje zinterpretowanie ciągu znaków wskazywanego przez **format** i wyprowadzenie do pliku identyfikowanego przez **file**, sekwencji znaków określonych łącznie przez **format** oraz argumenty skojarzone z parametrem **...** (wielokropek). Rezultatem jest liczba wyprowadzonych znaków.
Przyjmuje się, że ciąg wskazywany przez **format** składa się ze znaków oraz z wzorców konwersji. Wymaga się, aby każdemu wzorcowi konwersji odpowiadał jeden argument skojarzony z parametrem. Zinterpretowanie znaku występującego poza wzorcem konwersji powoduje wyprowadzenie go do pliku reprezentowanego przez **file**. Zinterpretowanie wzorca konwersji związanego z pewnym argumentem powoduje wyprowadzenie ciągu znaków, mającego postać literału reprezentującego wartość argumentu. Jeśli wyprowadzanie odbywa się do pliku identyfikowanego przez stdout, to wywołanie funkcji fprintf może być przedstawione w postaci:
printf(**format,a1,a2, ... ,an**)
Wzorzec konwersji może mieć postać %c, %s albo %d, w wyniku czego następuje wyprowadzenie danej reprezentowanej przez parametr jako pojedyńczego znaku, ciągu znaków albo liczby dziesiętnej.

Przykład
Sposób użycia funkcji printf przedstawiono na wydruku 2. Wykonanie przytoczonego programu powoduje wyprowadzenie napisu
JanB is 45 now

Funkcja fscanf

Naglowek: int fscanf(**file,format, ...** |
FILE **file**:
char **format**.

Wykonanie funkcji fscanf powoduje zinterpretowanie ciągu znakow wskazywanego przez **format**. wprowadzenie z pliku identyfikowanego przez **file** sekwencji znakow okreslonych łącznie przez **format** oraz argumenty skojarzone z parametrem ... (wielokropek), potraktowanie ich jako zapisow literalow zewnętrznych i przypisanie danych wskazywanych przez te literały, zmiennym wskazywanym przez argumenty skojarzone z parametrami. Rezultatem jest tu dana o wartosci równej liczbie wykonanych przypisan albo o wartosci EOF, w wypadku napotkania końca pliku.
Przyjmuje się, że ciąg wskazywany przez **format** składa się z odstępow, z innych znaków oraz z wzorców konwersji. Wymaga się, aby każdemu wzorcowi konwersji nie zawierającemu znaku (gwiazdka) odpowiadał jeden argument. Jeśli wprowadzanie odbywa się z pliku reprezentowanego przez stdin, to wywołanie funkcji może zostać przedstawione w postaci

scanf(**format,a1,a2, ... ,an**)

Zinterpretowanie odstępu powoduje zignorowanie odstępu w pliku identyfikowanym przez **file**. Zinterpretowanie znaku nie należącego do wzorca konwersji powoduje pominięcie takiego samego znaku w pliku identyfikowanym przez **file** (w wypadku niezgodności następuje zakończenie wykonywania funkcji). Zinterpretowanie wzorca konwersji powoduje potraktowanie kolejnego pola pliku jako zapisu literału zewnętrznego. Jeśli bezpośrednio po znaku " rozpoczynającym wzorzec konwersji występuje znak , to dana reprezentowana przez ten literał zostanie przypisana kolejnej zmiennej wskazywanej przez argument. W przeciwnym razie pole wejściowe zostanie zignorowane. Wzorzec konwersji może mieć postać %c, %s albo %d. Zinterpretowanie wzorca konwersji różnego od %c powoduje pominięcie w pliku identyfikowanym przez **file** najbliższego odstępu, a następnie stosownie do wypadku, uznanie za pole jednego znaku, ciągu znaków aż do odstępu albo ciągu znaków aż do znaku nie należącego do liczby. Interpretowanie wzorca konwersji odbywa się w taki sposób, że w wypadku znaku nie należącego do pola jest on cofany do pliku.
Wykonanie funkcji fscanf kończy się z chwilą zinterpretowania całego ciągu **format**, z chwilą wprowadzenia znaku niepoprawnego albo uniemożliwiającego wykonanie konwersji, albo z chwilą napotkania końca pliku. Rezultatem funkcji jest dana o wartości równej liczbie pomyślnie dokonanych przypisań, albo dana o wartości EOF w przypadku napotkania końca pliku.

Przykład
Sposób użycia funkcji scanf przedstawiono na wydruku 3. Wykonanie przytoczonego programu dla danych.

Jan Bielecki is 45 now

spowoduje wyprowadzenie napisu

Jan B is 45 now

Literatura

1. Bielecki J.: Język C - interpretacja standardu, WNT 1987

Wydruk 1

```
#include <stdio.h>

main()
{
  FILE *inp;
  int ch;

  inp = fopen("JANB.DOC"  "r");

  if(inp){
    while((ch = getc(inp)) != EOF)
    putc(ch,stdout);
    close(inp);
  } else
    putc('?',stderr);
}
```

Wydruk

```
#include <stdio.h>

main()
{
  printf("Jan%c  %s"  'B',45,"now");
}
```

Wydruk 3

```
#include <stdio.h>

main()
{
  char Initial;
  int Age;
  char Name[41];

  scanf("%s %c%*s is %d%*s",  Name,
                              &Initial,
                              &Age);
  printf("%s%c  %d now",  Name,
                          Initial,
                          Age);
}
```

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Connection Machine 2

Najnowszy komputer o architekturze określanej jako Hyper-Cube może składać się z 16384, 32768 lub 65536 procesorów. Został wyposażony w osiem kanałów we-wy o przepustowości 320 megabodów każdy. Procesory dzielą między sobą 512 MB pamięci operacyjnej.
Pamięć masowa komputera składa się z jednostek o pojemności 5 lub 10 GB, z których każda jest zbudowana z 39-ciu dysków magnetycznych. Dla zwiększenia bezpieczeństwa informacji dane są zapisywane równocześnie na kilku dyskach. Do każdego kanału we-wy można przyłączyć jedną jednostkę pamięci dyskowej lub terminal z monitorem kolorowym o wysokiej rozdzielczości.
CM-2 może wykonywać 2,5 miliarda operacji sumowania 32-bitowych liczb całkowitych w ciągu sekundy. Wydajność przy obliczeniach zmiennoprzecinkowych pojedynczej precyzji wynosi 3500 MFLOPS i 2500 MFLOPS przy obliczeniach w podwójnej precyzji.
Komputer jest wyposażony w optymalizujący kompilator języka FORTRAN.
Cena CM-2 wynosi, w zależności od konfiguracji, od jednego do pięciu milionów USD. Datamation **(an)**

Ploter ImageGraph

Ploter firmy Fujitsu jest produkowany w dwóch wersjach: model A dysponuje powierzchnią roboczą 21,6x27,9 cm, B - 27,9x43,2 cm. Dokładność wykonywania rysunku wynosi 0,25 mm, powtarzalność 0,1 mm, maksymalna szybkość kreślenia 25 cm/s. Ploter jest wyposażony w sześć pisaków i waży niecałe pięć kilogramów.
ImageGraph jest całkowicie kompatybilny z HP 7475A, akceptuje komendy wydawane w języku do zastosowań graficznych HP GL Hewlett-Packarda. Posiada wejście Centronics i RS232C oraz bufor danych (6 KB). Datamation **(an)**

Całostronicowy monitor dla PC

Superview firmy Epsilon Graphics Computation to 19-calowy, monochromatyczny monitor dla PC XT/AT pozwalający zobrazować pełną stronę maszynopisu (rozdzielczość 1280x960 punktów).
Monitor dostarczany jest łącznie z kablem interfejsu, sterownikiem i oprogramowaniem zawierającym procedury sterujące dla programów: GEM, Microsoft Windows, AutoCAD, Ventura Publisher, czy Page Maker.
Poprawne działanie wielu innych programów jest zapewnione dzięki pełnej kompatybilności ze standardem CGA.
Cena 2395 USD. Byte 11/87 **(as)**

Program do rozwiązywania zadań numerycznych

Solver-Q to program umożliwiający rozwiązanie tak skomplikowanych problemów numerycznych, jak np. układu tysiąca równań nieliniowych. Program zawiera edytor pozwalający na zapisywanie wyrażeń algebraicznych w postaci symbolicznej. Posiada także interfejs programowy umożliwiający współpracę z programem Lotus 1-2-3.
Solver-Q wykorzystuje technikę okien. Może być uruchamiany na komputerach typu PC XT/AT, wyposażonych w co najmniej 256 KB RAM. Do pakietu dołączany jest również program uczący. Byte 11/87 **(as)**

Drukarki Citizen

Firma Citizen (znana do niedawna głównie w USA) zdobywa sobie coraz większą popularność na rynku zachodnioeuropejskim. W 1986 roku sprzedano w Europie 305 tys. sztuk różnych modeli drukarek Citizena - jak na pierwszy rok aktywnej promocji - niezły wynik. W 1987 roku sprzedaż wzrosła ... o 100%. Przeważał model LSP 120D wypełniający lukę między drogimi drukarkami przeznaczonymi do zastosowań profesjonalnych i tanimi lecz prymitywnymi drukarkami do komputerów domowych. Na drugim miejscu znajdował się model MSP 15E (z tzw. szerokim wałkiem - 15 cali). Duży popyt na LSP 120D spowodował szybką reakcję firmy Citizen - od września 1987 uruchomiono produkcję tego modelu na terenie Anglii. Fabryka ma wytwarzać 30 tys. sztuk miesięcznie.
Drukarki firmy Citizen znalazły się już w ofercie niektórych polskich firm dostarczających sprzęt komputerowy obok produktów firmy Star i Epson. Inf. własna **(AJP)**

Pomoc dla 386

C-Executive to ładowany z pamięci ROM, wieloprogramowy system operacyjny czasu rzeczywistego. Opracowano wersje 12 różnych jednostek centralnych, w tym na Intel 8088, 8086, 80286, Zilog Z80, Motorola 6809, 68000 i 6802. System został napisany prawie całkowicie w języku C, za wyjątkiem podprogramów krytycznych ze względów czasowych takich jak np. obsługa przerwań czy obsługa kolejki procesów, które zostały napisane w języku maszynowym.
C-Executive w wersji 80386 może pracować również na komputerach osobistych, kompatybilnych z IBM PC wyposażonych w ten mikroprocesor. Z systemem dostarczany jest kompilator języka C, Assembler, Linker, program obsługi zbiorów bibliotecznych oraz opcjonalnie system zarządzania zbiorami CE-FILE. Byte 11/87 **(as)**

Najszybsza stacja graficzna

System Quadstation firmy BenchMark może obsługiwać do czterech użytkowników. Został zbudowany w oparciu o pracujący pod kontrolą systemu UNIX minikomputer BenchMark 32. Sercem komputera jest mikroprocesor NS 32332 (15 MHz) oraz opcjonalnie CLIPPER lub MC 68020.
Każdy użytkownik dysponuje terminalem wyposażonym w kartę graficzną BCG wykreślającą w ciągu sekundy do 110 milionów punktów. Karta ma dwa bufory pamięci ekranu mieszczące obraz o rozdzielczości 1280x1024 punkty; oferuje paletę 16,7 miliona barw, z których równocześnie można wybrać 256. CAD **(an)**

Tania karta dla LaserJet

1-2-4 to nie jest nowa wersja programu Lotus, lecz karta do rozbudowy pamięci dla drukarek LaserJet serii II firmy Hewlett-Packard, dostarczana przez Pacific Data Products. Karta ta może zawierać 1, 2 lub 4 MB dodatkowej pamięci o czasie dostępu 150 ns. W celu jej zainstalowania wystarczy podnieść pokrywę drukarki, odpowiednio ustawić mikroprzełączniki na karcie i włożyć ją do gniazda rozszerzenia pamięci. Cena karty 1-2-4 od 295 USD (bez układów), do 1595 USD za kartę z 4 MB pamięci RAM. Byte 11/87 **(as)**

Na początku lat osiemdziesiątych użytkownicy komputerów wywodzący się spoza kręgu zawodowych informatyków, a więc inżynierowie, naukowcy, architekci, projektanci zachęceni walorami użytkowymi mikrokomputerów postanowili wykorzystać je w prowadzonych pracach. Okazało się jednak, że popularne mikrokomputery nie spełniają pokładanych w nich nadziei: miały za małą pojemność pamięci operacyjnej, za wolne procesory, zbyt małą rozdzielczość obrazu przy pracy w trybie graficznym. Natomiast klasyczne minikomputery, które spełniały wymagania stawiane w pracach inżynierskich, były za drogie, by trafić na każde stanowisko robocze w pracowniach projektowych i komórkach badawczych.

WORKSTATIONS

Pierwsze próby wypełnienia luki między minikomputerami i popularnymi mikrokomputerami miały miejsce w latach 1981-1982, kiedy firmy Apollo Computer i Sun Microsystems zaprezentowały swoje stacje robocze ogólnego przeznaczenia: Domain DN 100 i Sun-1, przeznaczone głównie do zastosowań inżynierskich. Stacje te zostały wyposażone w szybkie procesory, dużą pamięć RAM, monitory o wysokiej rozdzielczości, pojemne pamięci dyskowe oraz interfejsy umożliwiające pracę w sieci.

Między 1983 i 1986 rokiem, firmy Apollo i Sun zanotowały szybki wzrost obrotów, stając się głównie poddostawcami urządzeń dla wytwórców, którzy oferowali kompletne systemy służące do projektowania układów elektronicznych, konstrukcji lotniczych, mechanicznych, jak też do opracowywania oprogramowania, badań nad sztuczną inteligencją, a także do zastosowań finansowych i wydawniczych.
Poszczególne modele stacji roboczych, chociaż pochodzące od różnych wytwórców, wykazują niespodziewanie wiele cech wspólnych. Typowa stacja ma 32 bitowy procesor, 4...8 MB pamięci operacyjnej, 40...80 MB pamięci dyskowej, monitor o bardzo wysokiej rozdzielczości (około miliona punktow na ekranie), szybkość przesyłania w sieci rzędu 10 megabodów oraz wieloprogramowy system operacyjny ze stronicowaniem i mechanizmami pamięci wirtualnej.
W stacjach roboczych przeważnie stosowane są mikroprocesory z rodziny Motorola 68000. Do tej grupy można zaliczyć modele firm Apollo, Sun oraz starsze wyroby firmy Hewlett-Packard. Są jednak wyjątki, np. stacje typu VAX firmy DEC i RT PC firmy IBM mają procesory opracowane przez producenta stacji. Pojawiły się także konstrukcje z procesorem Intel 80386.

Nowe modele stacji roboczych zapewniają przede wszystkim coraz większą moc przetwarzania. Czasem polega to na zastosowaniu tego samego mikroprocesora z szybszym zegarem. Inną drogą jest użycie mikroprocesora z tej samej rodziny, ale o większej mocy obliczeniowej. Przykładowo, komputer IBM PC AT (w zasadzie nie powinien być zaliczany do stacji roboczych, ale jest czasem stosowany w ten sposób) można zastąpić przez PS/2 Model 80 (z procesorem 80386), który może już być rozbudowany do pełnej konfiguracji stacji roboczej. Coraz częściej producenci stacji roboczych stosują procesory oparte na koncepcji RISC (zredukowanej listy rozkazów). Na przykład firma Sun zdecydowała się na architekturę typu RISC w swojej ostatniej serii Sun-4.

Również IBM PC RT ma mikroprocesor RISC ROMP, a seria 800 modelu HP 9000 bazuje na układach scalonych RISC Hewlett-Packard Precision Architecture. Nawet firma DEC, której stacje typu VAX były konstruowane z układów MicroVAX (odzwierciedlają one architekturę minikomputera VAX) ogłosiła, że rozważa zastosowanie koncepcji RISC do stworzenia nowej generacji jednostek centralnych, realizujących funkcje stacji roboczej.

Poszukiwania te wynikają ze wzrastającej konkurencji ze strony popularnych komputerów osobistych o coraz lepszych parametrach.

Szybkość działania jednostek centralnych komputerów osobistych charakteryzuje się podając częstotliwość pracy ich zegara. W przypadku stacji roboczych stosuje się pojęcie określanej w milionach liczby rozkazów wykonywanych w ciągu sekundy (tzw. MIPS-ów). Pierwsze modele stacji roboczych pracowały z szybkością około 1 MIPS. Wkrótce wielkość ta została podwojona. Ostatnie modele, np. Apollo Domain 4000, oferują już około 4 MIPS. Najnowszy Sun-4, o podwyższonych parametrach (koszt 40 tys. USD w minimalnej konfiguracji), osiąga szybkość 10 MIPS.

Cechą wyróżniającą stacje robocze spośród innych mikrokomputerów jest zdolność do szybkiego wykonywania operacji zmiennoprzecinkowych, niezbędnych przy przetwarzaniu obrazów lub przy obliczeniach naukowo-technicznych. Większość stacji ma w tym celu specjalne wyposażenie. W wypadku tańszych modeli, jest to koprocesor arytmetyczny, współpracujący z głównym procesorem; w droższych - stosowane są dodatkowe karty ze specjalizowanymi procesorami arytmetycznymi. Stacje robocze z mikroprocesorem 68020 przeważnie są wyposażane w koprocesor 68881, a konstruowane z zastosowaniem 80386 - w koprocesory 80287, 80387 (dostępny od drugiej połowy 1987 roku) lub nawet w specjalizowane układy firmy Weitek.

Bardzo istotnym elementem stacji roboczej z punktu widzenia użytkownika jest monitor o dużej przekątnej ekranu (najczęściej 19 cali). W droższych systemach stało się to już prawie standardem. Tak duża powierzchnia obrazowania pozwala na jednoczesną prezentację kilku okien programowych i wymaga użycia rozdzielczości rzędu 1024x1024 punktów. Na przykład w stacji Sun-3/50M wynosi ona 1152x900 , Apollo Domain serii 3000 oferuje 1024x800, a serii 4000 - 1280x1024 punktów. Systemy wywodzące się z komputerów osobistych, jak IBM PS/2 lub Mac II, mają standardową rozdzielczość od 640x480 do 720x400 punktów. Stosuje się zarówno monitory monochromatyczne, jak i kolorowe z wykorzystaniem 4, 6 lub 8 bitów na kod koloru. Stacje zapewniające bardzo dobre parametry graficzne wykorzystują nawet 24 bity na kod koloru.

Nawet najdoskonalszy sprzęt nie pracuje sam. Prawdziwą jego wartość określa dopiero oprogramowanie systemowe i użytkowe. W stacjach roboczych najczęściej jako system operacyjny są stosowane różne adaptacje systemu UNIX, łączące zazwyczaj elementy wersji Berkeley 4.2 z wersją System V. Niektórzy producenci wykorzystują także swoje własne systemy operacyjne. Na przykład firma Apollo stosowała początkowo swój własny produkt, Aegis OS, a firma DEC oferuje, oprócz UNIX-a, system VMS, mający bogatą historię w dziedzinie obliczeń naukowych. Podobnie Apple dostarcza swój własny system Mac OS, a TeleVideo - własną wersję systemu MS-DOS.

Połączenie kilku stacji roboczych w sieć jest celowe w przypadku wspólnego korzystania z kosztownych urządzeń peryferyjnych (np. plottery dużego formatu, drukarki, dyski optyczne lub stacje taśm magnetycznych) oraz gdy istnieje potrzeba szybkiego przesyłania zbiorów danych pomiędzy stacjami. Najczęściej stosuje się sieci typu Ethernet. Używane są także AppleTalk i Token-Ring.
W zakresie oprogramowania sieciowego firma Sun jest bliska uzyskania dla swojego protokołu NFS (Network File System) statusu standardu. Pozwala on na przesyłanie zbiorów i komunikatów pomiędzy różnymi stacjami w sieci. System Unix, używany przez większość stacji, również dysponuje protokołem sieciowym RFS (ang. Remote File Sharing). Firmy Apollo i DEC oferują także swoje własne systemy, odpowiednio NCS (Network Computing System) i DECnet.
Wielu producentów dostarcza, specjalnie dla systemów sieciowych, stacje robocze o rozbudowanej pamięci operacyjnej i masowej, dostosowane do pracy jako węzły sieci (file servers). Można tutaj wymienić np. Apollo, Sun czy Hewlett-Packard. IBM nie przewiduje dla swojej rodziny szczególnego wyróżniania jakiegoś komputera jako stacji zarządzającej, a Apple dostarcza tylko oprogramowanie realizujące takie funkcje na standardowym sprzęcie.

Opr. **Andrzej Skrzela**
wg Byte 10/87

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Dyski optyczne

W pamięciach masowych komputerów od lat wykorzystywane są zjawiska magnetyczne. Od pierwszych rozwiązań bębnowych do współczesnych Winchesterów nastąpił ogromny postęp technologiczny, jednak użytkownicy wciąż nie są w pełni usatysfakcjonowani - domagają się jeszcze większych pojemności i szybszego dostępu do informacji.

Zastosowanie metod optycznych do zapisu informacji rozpoczęło się od elektroakustyki. Zarejestrowanie dźwięku w postaci cyfrowej wymagało zarówno wielkiej pojemności, jak i dużej szybkości transmisji - okazało się więc, że płyty cyfrowe CD (Compact Disc) można z powodzeniem wykorzystywać jako pamięci masowe komputerów.

Klasyczny Compact Disc zawierający nagrania dźwiękowe nie różni się prawie od tzw. CD-ROM - pamięci optycznej tylko do odczytu. Budowa i zasada działania są nie mal identyczne.

Na metalizowanej płytce z tworzywa sztucznego wytłoczone są zagłębienia (tzw. pity), a zapisana w ten sposób informacja odczytywana jest skoncentrowaną wiązką światła laserowego odbijającą się (lub nie) od metalizowanej powierzchni. Różnice dotyczą jedynie sposobu kodowania danych. W przypadku dysków akustycznych stosowane są kody mające minimalizować szumy i zniekształcenia dźwięku powodowane przez błędy odczytu, natomiast mniej istotne jest, czy poszczególne bajty danych zostały odczytane prawidłowo. W CD-ROM trzeba odtworzyć pełną informację - kody korekcyjne są więc inne, zazwyczaj bardziej skomplikowane. Stosuje się też technikę przeplatania danych (ang. Interleaving) minimalizującą wpływ błędów grupowych (rys. 1).

Optyczne pamięci CD-ROM są już obecnie dostępne na rynku (w RFN cena ok. 3500 DM). Na dysku o średnicy 4,72 mieści się zazwyczaj 550 MB, co odpowiada ok. 300 tys. stron tekstu. Najczęściej na CD-ROM umieszczane są encyklopedie, słowniki, katalogi części i podzespołów, rozkłady jazdy i lotów. Prawnicy z krajów anglosaskich, w których obowiązuje tzw. prawo precedensowe, z niecierpliwością oczekują dysku CD-ROM zawierającego zbiór rozstrzygnięć prawnych, a konstruktorzy samochodowych systemów komputerowej nawigacji planują umieszczanie na dyskach optycznych map drogowych i planów miast.

Fakt, że cała informacja zawarta w CD-ROM jest zapisywana w procesie produkcji dysku - ogranicza jednak zakres możliwych zastosowań. Pełnowartościową pamięcią masową dla komputerów będzie można nazwać dopiero urządzenie umożliwiające wielokrotny zapis i odczyt.

Takich pamięci nie ma jeszcze na rynku, ale liczne egzemplarze prototypowe i laboratoryjne pozwalają przewidzieć, jakie będą ich cechy.

W dyskach optycznych z możliwością wielokrotnego zapisu stosowana jest warstwa specjalnego stopu, który pod wpływem temperatury i pola magnetycznego zmienia swoje właściwości (np. współczynnik odbicia światła lub kąt polaryzacji światła odbitego).

Zapisu na takiej warstwie dokonuje się metodą magnetyczno-optyczną: głowica zapisująca wytwarza pole magnetyczne o kierunku zależnym od zapisywanych danych, a silna wiązka laserowa rozgrzewa (ale nie niszczy) warstwę stopu.

Po jej zastygnięciu, na powierzchni dysku powstają punkty (pity), które w różnym stopniu odbijają światło. Odczyt następuje podobnie jak w CD-ROM.

Promień laserowy (znacznie słabszy niż przy zapisie) odbija się od warstwy stopu, a zmiany jego natężenia lub kąta polaryzacji odtwarzają zapisane informacje.

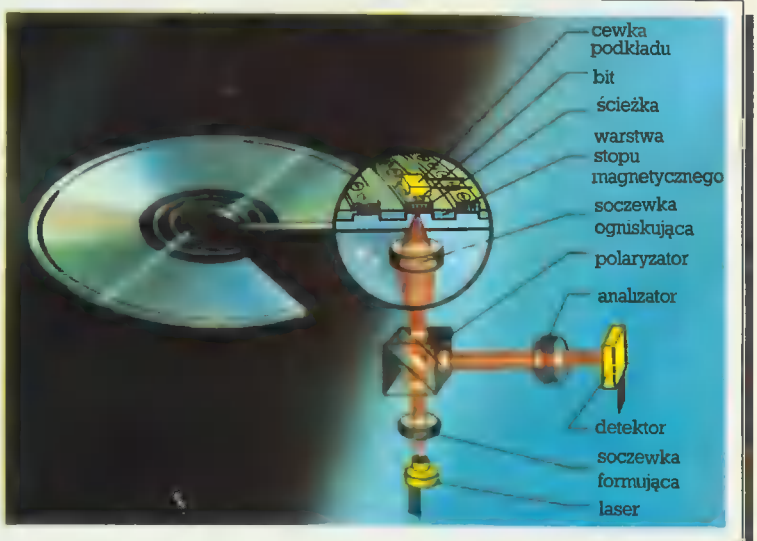
Obecnie największe perspektywy otwierają się przed pamięciami typu WORM (Write Once, Read Multiply), które umożliwiają zapisywanie (i dopisywanie) danych przez użytkownika. Takie systemy są już dostępne (ceny w RFN od 5 tys. DM) i oferują pojemności od 150 do 800 MB.

Zapis odbywa się przez wypalanie promieniem lasera (poprzez warstwę ochronną) dziur-kraterów w wartwie metalu pokrywającego podłoże z tworzywa sztucznego. Gęstość zapisu informacji na dysku WORM jest o rząd wielkości większa niż dla dysków magnetycznych - sięga 25 tys. bpi (bitów na cal). Odczyt jest taki sam jak w CD-ROM - promień lasera odbija się od metalizowanej powierzchni, ale jest pochłaniany w kraterach.

Dyski typu WORM stwarzają duże trudności nie tylko konstruktorom i technologom, ale przede wszystkim programistom tworzącym systemy operacyjne dla takich pamięci. Przeszkodą okazują się tablice alokacji plików FAT (ang. File Allocation Table), które nie mogą być dynamicznie modyfikowane, tak jak na dyskach magnetycznych. Trudne jest też dokonywanie zmian i aktualizacji w istniejących danych.

```
Ciąg danych:                    Optyczna_pamięć_masowa...

Zapis w matrycy:                O z a _ w
                                p n m m a
                                t a i a .
                                y _ ę s .
                                c p ć o .

Ciąg danych zapisywany na dysku:  Oza_wpnmmataia.y_ęs.cpćo.

Uszkodzony zapis na dysku:      Oza_wpn######ia.y_ęs.cpćo.

Uszkodzony zapis w matrycy:     O z a _ w
                                p n # # #
                                # # i a .
                                y _ ę s .
                                c p ć o .

Odczyt:                         Op#yczn#_pa#ięć_#asow#...
```

1. Metoda przeplatania.



2. Konstrukcja napędu dysku optycznego.

Nie powstał jeszcze jednolity, standardowy system operacyjny dla dysków optycznych. Stosowane są różne rozwiązania.
* Informacje o strukturze dysku są przechowywane na oddzielnej dyskietce i wykorzystywane tak, jak w przypadku dysków magnetycznych. Taka metoda powoduje jednak ogromną stratę szybkości, a zniszczenie informacji na dyskietce powoduje, że dane na dysku WORM stają się bezużyteczne.
* W trakcie inicjalizacji systemu tablica FAT jest wczytywana do pamięci RAM, a po zakończeniu pracy zapisywana na dysku. Jest to rozwiązanie bardzo efektywne, ale awaria zasilania lub wyłączenie komputera przed zapisem FAT prowadzi do utraty danych na dysku WORM.
* Po każdej aktualizacji danych na dysku zapisywana jest nowa ersja FAT (technika overhead). W przypadku często wykonywanych zapisów może się okazać, że więcej miejsca na dysku zajmują kolejne wersje FAT, niż właściwe dane.
* Zapisy podlegające aktualizacji są lokalizowane przez wskaźniki w tablicy alokacji. Nie trzeba tu powielać całych zbiorów; można także sięgać do poprzednich wersji danych i programów.

Zbiory danych istniejące już na dysku WORM mogą być zmieniane dwoma sposobami: ponownie zapisując cały plik po modyfikacji albo tylko jego zmienione fragmenty. Pierwsza metoda jest nieefektywna, a ponadto można ją stosować tylko w przypadku plików o niewielkich rozmiarach: nie sposób zaktualizować zbioru 150 MB na dysku mieszczącym 200 MB.
Najlepsza ze stosowanych obecnie metod zmieniania i uzupełniania danych na dysku WORM polega na umieszczeniu w każdej jednostce alokacji pola adresowego, które identyfikuje położenie ewentualnej modyfikacji tego fragmentu danych. Wskazana w ten sposób jednostka alokacji może zawierać wskazanie na kolejną wersję itd. W metodzie tej wykorzystanie pojemności dysku WORM zbliża się do maksymalnego. Ponadto można prześledzić historię wykonywanych zmian.
Do czasu opracowania systemu operacyjnego dla dysków WORM i pojawienia się na rynku dysków optycznych z możliwością wielokrotnego zapisu pozycja magnetycznych pamięci masowych nie wydaje się zagrożona. Mimo to w specyficznych zastosowaniach wymagających dużej pojemności i szybkiego dostępu dyski optyczne już dziś stanowią najlepsze ze znanych rozwiązań.

Opr. Janusz Wrześniak Computer Persoenlich 1/87

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

ELEKTRONICZNA POCZTA

Na polskim rynku komputerowym coraz częściej pojawiają się zestawy działające po kontrolą systemu operacyjnego UNIX. System ten może zostac wykorzystany do stworzenia sieci poczty elektronicznej w biurze, fabryce lub uczelni.
Przesyłanie informacji odbywa się w trybie natychmiastowym (poleceniem **write**) lub pośrednio przez system zarządzający pocztą (poleceniem **mail**).
Polecenie write umożliwia przesłanie dowolnej informacji do każdego użytkownika systemu, ktory jest aktualnie zarejestrowany (**login**). Polecenie to ma postać:
write user tty
user jest tu identyfikatorem użytkownika, a **tty** oznacza nazwę wykorzystywanego terminalu. Drugi parametr może nie być okreslony, jeżeli jednocześnie nie pracują dwaj użytkownicy o tym samym identyfikatorze. Poniższy przykład przedstawia reakcję systemu na próbę przesłania wiadomości od użytkownika nie zarejestrowanego. We wszystkich przykładach znak $ oznacza gotowość systemu operacyjnego do przyjmowania poleceń (ang. prompt).
$ write Jacek
Jacek is not logged on.
$
Jeżeli Jacek został wcześniej zarejestrowany przez system, a wiadomość wysłałby Jurek z terminalu **tty19**, to po sygnale dźwiękowym na ekranie monitora Jacka pojawi się komunikat:

Message from Jurek tty19...

Niektóre odmiany systemu UNIX, na przykład wersja V, wyświetlają dodatkowo aktualną datę i godzinę. Po zainicjowaniu łącznosci poleceniem **write**, system oczekuje na wprowadzenie z klawiatury ciągu znaków. Mogą to być również znaki powrotu karetki CR i nowej linii LF. Jednoczesne wciśnięcie klawiszy CTRL i d kończy wprowadzanie informacji i powoduje przesłanie tekstu do adresata. Wśrod użytkowników systemu UNIX przyjął się zwyczaj pisania na końcu przesyłanego tekstu znaków **-o**, oznaczających oczekiwanie na odpowiedź (z ang. over). Znaki **-oo** oznaczają chęć zakończenia konwersji (z ang. over and out).

Pojawienie się komunikatów pochodzących od innych użytkowników może być w niektórych przypadkach niewskazane lub wręcz szkodliwe. Istnieje możliwość zablokowania komunikacji z innymi użytkownikami przez wydanie polecenia **mesg n**. Poleceniem przeciwnym (odblokowującym odbiór) jest **mesg y**. Użytkownik może w każdej chwili sprawdzić czy odbiór jest zablokowamy:
$mesg
is n
$
Jeżeli odbiór nie jest zablokowany i na ekranie, w trakcie pracy (np. podczas edycji), pojawi się wiadomość od innego użytkownika, to wciskając klawisze CTRL l można przywrocić poprzedni stan ekranu. Próba przesłania wiadomości do użytkownika, który zablokował odbior powoduje pojawienie się komunikatu:
$ write Jacek
Permision denied.
$
Polecenie **mail** umożliwia przesłanie wiadomości użytkownikom zarejestrowanym i nie zarejestrowanym aktualnie przez system. Wysłana informacja nie jest przekazywana bezpośrednio do adresata, lecz zostaje zapisana przez system operacyjny na dysku i może być odczytana w dowolnym momencie. Rozpoczęcie redagowania listu następuje po wydaniu polecenia:

mail user

Od tego momentu system operacyjny traktuje - do czasu wciśnięcia klawiszy CTRL i d - wszystkie wprowadzone znaki jako treśc listu. Parametrem **user**, podobnie jak dla polecenia **write**, może być ciąg oddzielonych spacjami identyfikatorów użytkownikow. Wiadomosc zostanie przesłana do wszystkich wymienionych adresatów. Co pewien czas, jak też bezpośrednio po zarejestrowaniu nowego adresata, poczta każdego użytkownika jest przeglądana. Jeżeli pojawi się w niej nowa pozycja, to użytkownik otrzyma komunikat:
you have mail
Odbiorca może przejrzeć otrzymane listy, wydając polecenie:

mail

Po wyświetleniu każdej pozycji system zgłasza się, pisząc na ekranie znak **?** i oczekuje na dyspozycje (tab. 1).
Po przejrzeniu wszystkich listów system ponownie zgłasza się znakiem **$**.
W przypadku, gdy elektroniczna skrzynka pocztowa jest pusta, na polecenie **mail** system odpowiada komunikatem:

No mail.

Podstawowe polecenie elektronicznej poczty systemu operacyjnego UNIX

Dyspozycja	Opis
RETURN	Pozostawienie pozycji w skrzynce, wyświetlenie następnej
d	Usunięcie pozycji, wyświetlenie następnej
s	Zapisanie pozycji na dysku w zbiorze o określonej nazwie
q	Zakończenie przeglądania listów
?	Wyświetlenie listy dyspozycji (help)

Istnieje możliwość wcześniejszego przygotowania listu, na przykład za pomocą edytora tekstów i przesłania go w wybranym momencie. W tym celu należy skorzystać z podstawowej właściwości systemu UNIX, zamieniając rodzaj wejścia: klawiaturę zastąpić zbiorem dyskowym. Przedstawione poniżej polecenie powoduje przesłanie wiadomości zapisanej w zbiorze list do użytkownika o identyfikatorze **Ola**.

Mail Ola<list

Komputery zarządzane przez system operacyjny UNIX mogą być łączone ze sobą poprzez łącza telefoniczne i modemy. Komunikacja między użytkownikami różnych komputerów odbywa się za pomocą polecenia **mail**, na zasadach opisanych powyżej. Jedyna różnica polega na tym, że przed identyfikatorem użytkownika należy dopisać identyfikator komputera oraz znak **!**.

Marek Mizeracki

Nowy komputer przenośny HP

Z wyglądu przypomina zwykłą teczkę, ale tak naprawdę jest nowym, przenośnym komputerem osobistym Vectra kompatybilnym z IBM PC. Jego wymiary wynoszą: 42x35x9 cm, waga około 8 kg łącznie z bateriami zasilającymi przystosowanymi do ładowania. Oferowana jest także wersja wyposażona w sztywny dysk. Płaski 12 calowy ekran monitora ciekłokrystalicznego został umieszczony w górnej pokrywie obudowy. Pracuje on w trybie CGA, lecz z rozdzielczością 640x400 punktów i pozwala na wyświetlenie 25 linii po 80 znaków. Możliwe jest odłączenie standardowego wyświetlacza i dołączenie zewnętrznego monitora. Portable Vectra wykorzystuje mikroprocesor 8086 wykonany w technologii CMOS. Pracuje on z zegarem 7,16 MHz; możliwe jest stosowanie koprocesora 8087. Pamięć RAM (640 KB) może zostać rozszerzona do 6 MB (4 MB w wersji z wbudowanym dyskiem sztywnym). Klawiatura składa się z 92 przycisków.
Komputer Portable Vectra dostarczany jest w dwóch wersjach: pierwsza ma dwie stacje 3,5 calowych dysków elastycznych o pojemności 1,44 MB, a druga jedną stację dysków elastycznych i 20 MB dysk sztywny. Czas pracy, bez zewnętrznego zasilania, wynosi 10 h w pierwszej wersji i 4 h w drugiej.
Standardowo, Portable Vectra wyposażony jest tylko w interfejs równoległy (nie posiada interfejsu szeregowego dostarczanego na życzenie klienta). Z komputerem można zakupić system operacyjny Vectra DOS 3.2, będący funkcjonalnym odpowiednikiem systemu DOS 3.3. Cena w USA od 2495 USD. Byte 11/87 **(as)**

Przełom w archiwizacji danych?

Za pomocą techniki zapisu podobnej do stosowanej w popularnych videomagnetofonach, firma Emerald ma nadzieję na stworzenie poważnej konkurencji dla klasycznych pamięci taśmowych. Jej urządzenie oparte o technologię VAST (ang. virtual archive storage technology) pozwala na przechowywanie kopii zawartości dysków sztywnych, dysków optycznych oraz innych nośników pamięci o dużej pojemności. Szybkość transmisji wynosi 15 MB na minutę. Kasety używane w urządzeniach typu VAST pozwalają na zapamiętanie 250, 500, 1000 i 2200 MB informacji (odpowiada to pojemności ok. 6 tys. dyskietek). Dla tych użytkowników, którym podane wielkości jeszcze wydają się niewystarczające, opracowano sposób łączenia urządzeń typu VAST szeregowo z możliwością uzyskania pojemności przekraczającej 15 GB.
W urządzeniach VAST zastosowano technikę korekcji błędów pozwalającą na odtworzenie, w bloku 1024-bajtowym, błędnych ciągów o długości do 264 bajtów oraz na dodatkową korekcję dalszych 80 błędów losowych.
Razem z urządzeniem dostarczane jest oprogramowanie pozwalające na kopiowanie na kasetę zawartości wcześniej wybranych zbiorów, katalogów lub całego nośnika.
Możliwe jest również przegrywanie tylko tych zbiorów, które zostały ostatnio zmodyfikowane.
Urządzenie dostarczane jest łącznie z kablem interfejsu i kartą sterownika. Cena - jak na razie - od 6995 USD. Byte 11/87 **(as)**

Generator liczb pseudolosowych

Podstawowym narzędziem wykorzystywanym przy symulacji i analizie rzeczywistych bądź teoretycznych problemów numerycznych jest generator liczb pseudolosowych. Algorytmy uzyskania ciągów liczb o złożonych charakterystykach czasowych wymagają szybkości i dokładności obliczeń, są więc często realizowane w postaci programów komputerowych.

Algorytm tworzenia liczb pseudolosowych należy do klasy liniowych generatorów mieszanych i został zastosowany przez firmę Texas Instruments [2]. Opisuje go równanie o postaci:
x[n+1]= (x[n]*24298+99991)mod 199017
rand=x[n+1]/199017,0
Pierwotnie algorytm generacji liczb pseudolosowych zrealizowano w języku C, stosując 32-bitową arytmetykę liczb całkowitych (wydruk 1). Okazało się jednak, że wyniki pracy programu nie są poprawne. Przyczyną było przekroczenie zakresu liczb typu **unsigned long** (32 bity na liczbę bez znaku), ponieważ na czterech bajtach można zapisać cyfry z przedziału 0...4294267295, natomiast największa wartość jaka może wystąpić podczas obliczeń wynosi:
24298*199016+99991=4835790756
Zrezygnowano więc z arytmetyki liczb całkowitych i dla reprezentacji zmiennych zastosowano typ **double** (długi zmiennoprzecinkowy). Otrzymane w ten sposób wyniki (program 2) były poprawne, ale w związku z koniecznością użycia funcji **fmod** (operacja modulo na liczbach zmiennoprzecinkowych - **float**), zwiększył się czas obliczeń. Dla zwiększenia szybkości działania programu algorytm zapisano w języku maszynowym (wydruk 3), korzystając z rozkazów odwołujących się do koprocesora arytmetycznego. Program ten składa się z dwóch procedur: **m_rand** i **m_srand**. Procedura **m_srand** służy do inicjalizacji generatora liczb pseudolosowych - argument funkcji (liczba typu **double**) jest wartością początkową serii generowanych liczb. Procedura **m_rand** jest właściwym generatorem i zwraca liczbę typu **double** z zakresu (0,1). Przed użyciem funkcji **m_rand** należy zadeklarować typ zwracanej przez nią wartości jako **double** dyrektywą preprocesora:
double cdecl m_rand();
albo deklaracją typu
extern double m_rand();.
Po przeprowadzeniu podstawowych testów właściwości statystycznych przedstawionego generatora (test zgodności wartości oczekiwanej średniej arytmetycznej i test zgodności wartości oczekiwanej średniej kwadratów) okazało się, że otrzymane wyniki pokrywały się z wynikami testów przedstawionymi w pracy A.Podgórskiego [1].

Maciej Z. Markowski

Literatura:
[1] Andrzej Podgórski, Rozprawa doktorska: Zastosowanie procesora cyfrowego do wspomagania pomiarów cieplnych, Warszawa '83, Politechnika Warszawska.
[2] TI Programmable 58/59, Master Library, Applied Satistics, Texas Instruments Inc. Dallas-Texas '87.

```
double
rdm (seed)
double *seed;
{
return (*seed=
 fmod((24298.0* (*seed)+99991.0),199017.0))/199017.0;
}
```

```
double
rdm (seed)
long *seed;
{
return (double)(*seed=
 ((24298L* (*seed)+99991L)%199017L))/199017.0;
}
```

```
_TEXT           SEGMENT byte public 'CODE'
_TEXT           ENDS
_DATA           SEGMENT word public     'DATA'
_DATA           ENDS
DGROUP          GROUP   _DATA
                ASSUME  CS: _TEXT,ds: DGROUP
                public  _m_rand
                public  _m_srand

_DATA           SEGMENT
wynik           dq      ?
mnozna          dd      24298
skladnik                dd      99991
dzielnik                dd      199017
m_seed          dq      ?
_DATA           ENDS

_TEXT           SEGMENT
_m_rand proc    near
                fild    dzielnik            ;zaladuj dzielnik na szczyt stosu
                fild    [illegible]         ;zaladuj [illegible] na szczyt stosu
                fmul    m_seed              ;pomnoz ja przez wynik poprzed-
                                            ;niego wywolania procedury
                                            ;- rezultat tej operacji zostaje
                                            ;zapamietany na szczycie stosu
                fiadd   skladnik            ;dodaj skladnik do szczytu stosu
                fprem                       ;operacja modulo
                fst     m_seed              ;zapamietaj posiew dla
                                            ;nastepnego wywolania procedury
                fdiv ST,ST(1)               ;dziel szczyt stosu przez
                                            ;dzielnik znajdujacy [illegible] ST(1)
                fstp    wynik               ;zapamietaj wynik jako
                                            ;liczbe zmiennoprzecinkowa
                ffree   ST(0)               ;zwolnienie rejestru 8087
                mov     ax, offset DGROUP:wynik
                                            ;przekazanie wyniku do procedury
                                            ;nadrzednej
                ret
_m_rand         endp

_m_srand        proc    near
                push    bp
                mov     bp,sp
                fld     qword ptr [bp+4]    ;pobieranie liczby [illegible] stosu
programu
                fstp    m_seed              ;i zapamietanie jej
                pop     bp
                ret
_m_srand        endp

_TEXT ENDS
                end
```

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

(3) Trzeci wymiar

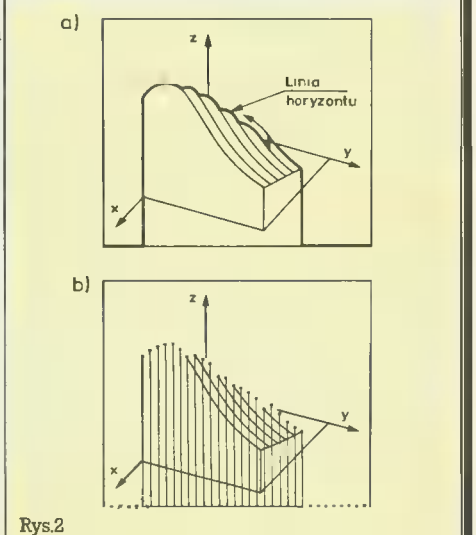
W poprzednich odcinkach cyklu zwróciliśmy uwagę na tzw. rysunek kreskowy. Rysunek taki generowany jest wyłącznie za pomocą układu nie zlewających się linii, tworzących najczęściej regularną siatkę. Technika ta w literaturze anglojęzycznej określana jest terminem wire-frame. Kolejną czynnością przy tworzeniu rysunku trójwymiarowego jest usuwanie linii, które są zasłaniane (ang. hidden lines removing). W rzeczywistości oczka sieci tworzą powierzchnie nieprzezroczyste, natomiast w przedstawionych w poprzednim artykule algorytmach rzutowania kreślone są wszystkie linie, zarówno widoczne jak i nie.

Aby sobie lepiej uświadomić, co to znaczy, że pewne obiekty są niewidoczne, można posłużyć się wykorzystywanym w pierwszym odcinku cyklu aparatem telefonicznym. Można założyć, że odpowiedź na to pytanie brzmi: widoczna jest część przednia - czyli bliższa, a niewidoczna tylna - czyli dalsza. Jest to oczywiście stwierdzenie błędne, bo po pierwsze, co oznacza - w odniesieniu do powierzchni - stwierdzenie „bliższa" lub „dalsza", a po drugie - to co leży dalej nie musi być wcale zasłonięte. Można więc powiedzieć inaczej: część przednia - czyli ta, która jest skierowana w stronę patrzącego, zaś część tylna - skierowana w stronę przeciwną. Badając w każdym punkcie wektor normalny do powierzchni, czyli wektor do niej prostopadły i odpowiednio skierowany, można łatwo sformalizować taki sposób opisu. W przypadku analizy brył powierzchnia skierowana w stronę przeciwną do obserwatora jest rzeczywiście niewidoczna. Jednakże stwierdzenie odwrotne: że powierzchnia zwrócona w stronę obserwatora jest widoczna - nie zawsze jest prawdziwe. Na przykład w analizowanym przypadku, odłożona słuchawka zasłania fragmenty widełek skierowane w stronę patrzącego. Kryterium tego typu może więc służyć tylko do wstępnej selekcji powierzchni widocznych i niewidocznych. Zgodnie z pierwszą koncepcją, obiekt - a ściślej jego konkretny punkt - będzie zasłaniany, jeśli na linii między tym punktem a okiem obserwatora znajdzie się inny nieprzezroczysty obiekt. Takie sformułowanie oznacza konieczność sprawdzania dla każdego punktu relacji przesłaniania a wszystkimi pozostałymi punktami. Zadanie takie jest wyjątkowo czasochłonne. Ustawienie wszystkich punktów od najdalszego do najbliższego (punkty leżące dalej byłyby wygaszane przez punkty leżące bliżej), czyli sortowanie zamiast sprawdzania relacji, nie przynosi zasadniczych zysków czasowych. Geometria analizowanej powierzchni nie jest jednak opisana przez ciągły zbiór punktów, ale przez dyskretną siatkę, której oczka po zrzutowaniu na ekran monitora obejmują najczęściej obszar rzędu kilkudziesięciu punktów obrazu. Wynika stąd, że sortować można grupowo. Wystarczy więc sprawdzać relację bliższy-dalszy nie w odniesieniu do punktów, lecz do oczek siatki.

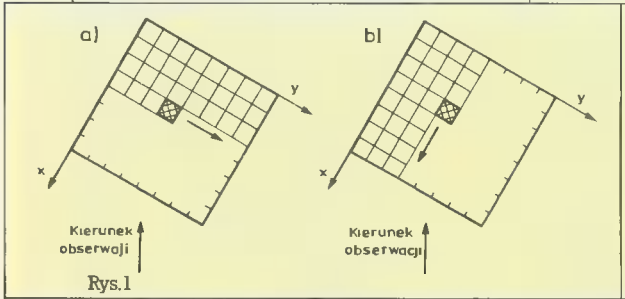
czarnym w białej obwódce. Powoduje to wygaszanie wszystkich, zapalonych uprzednio w obszarze ramki, punktów na ekranie. Algorytm tego typu stosowany jest w popularnym pakiecie graficznym ENERGRAPHICS. Jego wadą jest to, że obraz utworzony na ekranie w grafice rastrowej może być skopiowany jedynie na drukarce graficznej, o ile równolegle nie był tworzony - za pomocą innego algorytmu - rysunek w grafice wektorowej. O wiele większym zainteresowaniem, ze względu na możliwość wykorzystania plotera lub instrukcji emulujących jego działanie na monitorze, cieszą się algorytmy usuwania linii niewidocznych, pracujące w grafice wektorowej, a więc kreślące linie od punktu do punktu. Fakt, że w profesjonalnym sprzęcie powszechnego użytku rysunek na ekranie nadal składa się z punktów, jest już cechą sterownika obrazu. Istotny jest natomiast sposób generowania danych. Nasuwa się więc następująca metoda. Ponieważ rysunek składa się z linii, należy dla każdej z nich sprawdzić, czy jej końce leżą, po zrzutowaniu, wewnątrz kolejnych oczek oraz czy - w porównaniu z nimi - są położone bliżej obserwatora. W przypadku, gdy zachodzi to tylko dla jednego z końców odcinka, należy określić, z którą krawędzią następuje przecięcie i obliczyć współrzędne dla tego punktu. Na koniec pozostaje wykreślić odcinek o pełnej długości lub po skróceniu, albo też przejść bezpośrednio do analizy następnej linii. Algorytm ten jest prosty, ale mało efektywny. Modyfikacja, jaką należy zastosować, zbliżona jest do tej, która została opisana w grafice rastrowej. Analogicznie - na podstawie przesłanek wynikających z regularności siatki - można a priori określić kolejność analizowanych wektorów, tym razem poczynając od najbliższego. Ponieważ nie ma możliwości wymazywania linii niewidocznych, definiuje się tzw. linię horyzontu, poniżej której nie można już nic wykreślić. A więc modyfikacja w głównej mierze polega, na znajdowaniu przecięć kolejnych linii nie z bokami oczek, a z linią horyzontu. Linia ta zapamiętana może być na dwa sposoby. Pierwszy, pokazany na rysunku 2a, polega na zapisaniu w macierzy współrzędnych jej kolejnych węzłów. Podejście takie ma tę wadę, że linia ta ulega stałej modyfikacji. Pewne punkty do niej dochodzą,

stru. Wiąże się to z pewnymi błędami, jednak ich wartość jest mniejsza od odstępu między kolumnami. Gdy przyjmuje się dla plotera odstęp rzędu 0,1 mm, błąd taki nie spowoduje żadnych komplikacji, a jakość rysunku będzie całkowicie zadowalająca. Istnieje kilka modyfikacji tych metod, różniących się wyborem kolejności analizy wektorów. Pokazane zostały na rysunkach 3abc. Wersję z rysunku 3b zastosowano w dość rozpowszechnionym na rynku amerykańskim, a także i europejskim, profesjonalnym pakiecie graficznym ogólnego przeznaczenia: DISSPLA. Istnieją także jego adaptacje na komputery klasy IBM PC. Na uwagę zasługuje natomiast wersja pokazana na rysunku 3c, zastosowana w pakiecie SMALLPLOT pierwotnie dla PDP-11, która - mimo że ogranicza poważnie wybór kierunków rzutowania - jest znacznie szybsza, ze względu na zmniejszenie rozmiarów linii horyzontu. Przedstawiony algorytm oraz jego modyfikacje prowadzą do wykreślenia, w wypadku wykresu funkcji **f=f(x,y)**, jedynie jego górnej strony. Chcąc kreślić także dolną stronę, która w pewnych ujęciach może być widoczna, należy zdefiniować analogiczną dolną linię horyzontu. W ten sposób obszar zasłaniający jest zamknięty pomiędzy dolną i górną linią horyzontu. Algorytm ten nie ma zastosowania w przypadku brył. Tutaj zamiast linii horyzontu, definiuje się wielospójne, ograniczone wielokątami, obszary zasłaniające. Mimo większej złożoności ich kontrolowania, szczególnie wykonywania operacji dodawania, mnożenia i odejmowania pól, metoda postępowania pozostaje ta sama. Opracowania wymaga jedynie efektywna procedura sprawdzająca czy dany punkt znajduje się wewnątrz, czy na zewnątrz zakazanego obszaru. Przedstawione w dużym skrócie algorytmy usuwania linii zasłanianych należą (razem z technikami rzutowania i poziomicowania) do kanonu zagadnień grafiki trójwymiarowej.

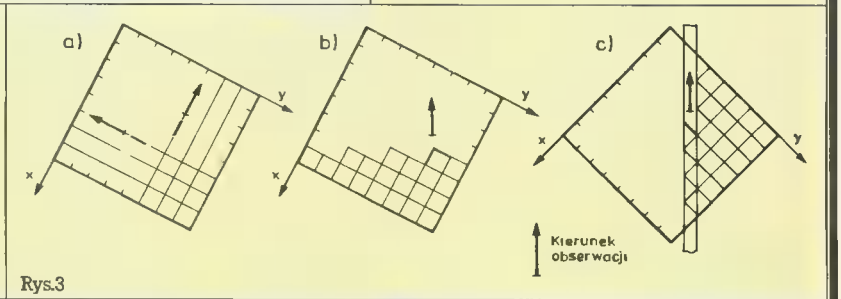
Drugi etap tworzenia rysunku trójwymiarowego stanowi nadawanie obiektom ich rzeczywistego



Rys.2



Rys.1



Rys.3

Procedura taka jest szczególnie prosta w przypadku regularnej siatki prostokątnej, dyskretyzującej wykres funkcji **f=f(x,y)**. Ze względu na jednakową orientację oczek także po zrzutowaniu, łatwo przewidzieć wymaganą kolejność ich analizy (rys. 1a i b). Zgodna jest ona z kolejnymi kolumnami lub wierszami siatki (licząc od najdalej leżącego oczka). Mimo że nie jest to kolejność od najdalszego do najbliższego, sortowanie tego typu daje pożądany efekt. Jeśli stosuje się grafikę czarno-białą, oczka siatki wyświetlane są w kolorze

pewne odchodzą, a ich całkowita liczba nie jest stała. Podejście alternatywne zobrazowane jest na rysunku 2b. Nawiązuje ono w pewnym sensie do metod grafiki rastrowej. Linia horyzontu zostaje tu zamieniona na ciąg punktów, których liczba jest równa liczbie kolumn w rastrze odpowiednio monitora lub plotera. Przy takim podejściu dokładnie wiadomo, między którymi kolumnami rastru kreślony jest kolejny wektor, a tym samym, gdzie zajść może ewentualne przecięcie. Oczywiście jego współrzędne sprowadzane są do kolumn ra-

wyglądu (ang. rendering), polegające na cieniowaniu, śledzeniu przebiegu światła i nakładaniu faktury. Aby uzyskać rysunek tej jakości, wymagane jest stosowanie sterownika obrazu o rozdzielczości co najmniej 640 na 480 punktów, 256 barwach używanych jednocześnie na ekranie z palety 4096 możliwych i odpowiedniej jakości monitora. Cieniowanie wiąże się z wypełnieniem siatkowej struktury obiektu barwą zależną od typu oświetlenia (fot. 1). Oświetlić można bowiem zarówno jednym, jak i kilkoma źródłami światła, światłem

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

rozproszonym albo punktowym. Kolor, jaki otrzyma dany fragment powierzchni, zależy od jego odległości od źródła światła (zawartość szarości), jak również od kierunku padania i odbicia oraz od położenia obserwatora (zawartości bieli). Jak się łatwo zorientować, analiza tego typu nie jest skomplikowana, a jedynie czasochłonna. Drugi etap - to badanie przebiegu promieni dla określenia położenia cieni jednych obiektów na drugich, załamania światła na obiektach przezroczystych oraz rozproszenia na powierzchniach rozpraszających. Nim zostanie omówiony etap trzeci trzeba przypomnieć, że w trakcie cieniowania brane były pod uwagę jedynie struktury geometryczne, bez uwzględnienia wyglądu powierzchni. Dokonuje się tego na etapie nakładania faktury. Zagadnienie to jest proste tylko w przypadku powierzchni rozwijanych. Tak jest, gdy nakładane są różne elewacje na ściany budynków projektowanego osiedla. Jeśli trzeba jednak wyświetlić np. znak reklamowy danej firmy owinięty dookoła kuli ziemskiej, staje się to już skomplikowane.

Robert Mołdach

Literatura:

J.D.Foley, A.Van Dam, Fundamentas of Interacitive Comuter Graphics, Addison-Wesley Pudling Company 1982.

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

Oferujemy do sprzedaży:

- mikrokomputery IBM PC XT/AT, Personal System/2, oraz 32–bitowe
- mikrokomputery AMSTRAD–SCHNAIDER
- urządzenia peryferyjne: drukarki, stacje dysków 3" i 5,25", dyski twarde, monitory, terminale, plotery, streamery i in.
- oprogramowanie użytkowe
- magnetowidy, kamery, monitory, kasety magnetowidowe.

Udzielamy gwarancji, zapewniamy serwis pogwarancyjny i materiały eksploatacyjne.

Spółdzielnia Pracy UNICUM
Dział Komputerów
00-504 Warszawa 15
skr. pocz. 20
Tel.49-56-66

unicum

Czy masz już te numery mikroklanu?

☒ **Wybieraj skreślając kratkę!**

☐ **Wpisz czytelnie liczbę!**

☐ **Zamawiam** 3 6 12 **kolejnych numerów mikroklanu**

od numeru ☐ /88.

Każdy numer po ☐ egzemplarzy

☐ Zamawiam następujące numery **mikroklanu**:

Nr.	2	1	9	8	7	6	5	4	3	2	1	3	2	1
Rok	88	88	87	87	87	87	87	87	87	87	87	86	86	86
Egz.	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Elektroniczne deski kreślarskie

Od czasu, gdy mikrokomputery coraz częściej wykorzystywane są do wspomagania prac projektowych można zaobserwować ponowny wzrost zainteresowania urządzeniami do dygitalizacji. Urządzenia opisywane w artykule (ang. graphics tablets) nazwaliśmy elektronicznymi deskami kreślarskimi, gdyż sposób posługiwania się nimi bardzo przypomina pracę kreślarza.

Porównanie własności kilku typów elektronicznych desek kreslarskich

	KoalaPad+ Touch Tablet	VersaWriter	Summa Sketch 1201	IS/ONE Series Tablet	Digi-Pad Type 5A Model 1111A	GP-7 Grafbar Mark II Sonic Digitizer	Pencept Penpa Model 320
Cena $	163	299	599	895 (przewodowy) 995(bezprzewodowy)	995 (wskaznik) 1265(16 przycisków celownik)	1315(wskaznik) 1485(4 przyciskow celownik)	1495
Wymiary zewnętrzne (cale)	6*8	12*13,5	16*16	16*15	16*16	bez obudowy	15*15
Obszar pracy	4.5*4.5	8.5*11	12*12	12*12	11*11	26*20	11*11
Urządzenie wskazujące							
Pisak	—	—	—	Tak	—	—	Tak
Wskaźnik	Tak	—	Tak	Tak	Tak	Tak	—
Celownik	—	—	Tak	Tak	Tak	Tak	(opcja)
Inne	Tak	ramię mech.	—	—	—	—	—
Oprogramowanie	PC Illustrator	VersaWriter Drawing System	brak	IS/Template (sterowanie okna i myszy Microsft)	brak	brak	sterowanie myszy programy graficzne i narzędziowe
Wejście	Game	Game	Szeregowe	Szeregowe	Szeregowe	Szeregowe	Specjalna plytka

Elektroniczne deski zapewniają większą rozdzielczość (rzędu 200 punktów na cal) i większą precyzję wskazań niż mysz. O ile mysz pozwala sterować położeniem kursora na ekranie to w przypadku desek elektronicznych możliwa jest także dygitalizacja rysunków.
Do wskazywania punktów na płaszczyźnie roboczej deski stosowane są celowniki, wskaźniki, lub pisaki. Służą one do zamiany przemieszczeń analogowych (ruchu ręki) na formę cyfrową (kod) wykorzystywany w programach.
W celowniku umieszczona jest soczewka i układ optyczny pozwalający precyzyjnie wskazać punkt. W obudowie celownika może też zostać umieszczone do kilkunastu przycisków uruchamiających funkcje związane z obsługą programu typu CAD - bez konieczności korzystania z klawiatury.
Wskaźnik przypomina kształtem pisak, nie zostawia jednak śladu prowadzonych linii.
Większość urządzeń wskazujących zasilana jest poprzez przewód łączący je z deską. Stosowane są jednak również wskaźniki z zasilaniem bateryjnym.
Niektórzy użytkownicy programów wspomagających prace projektowe stosują deskę kreślarską oraz pióro świetlne. Rysunki tworzone są przez nich na desce kreślarskiej, a następnie po przeniesieniu na ekran, poprawki i modyfikacje wykonywane są za pomocą pióra świetlnego.
Dla elektronicznych desek kreślarskich stosuje się kilka metod przetwarzania ruchu na postać cyfrową. W układzie pokazanym na rysunku 1 powierzchnia deski kreślarskiej spełnia rolę anteny, złożonej z sieci cienkich

☐ Dla odbiorców indywidualnych

IMIĘ NAZWISKO

ULICA GMINA

KOD POCZTA-MIEJSCOWOSC

Zamówione egzemplarze proszę przesyłać za zaliczeniem pocztowym, rownym cenie czasopisma (porto opłaca nadawca)

PODPIS

Wyślij pod adres:
Zakład Kolportażu
Wydawnictwa NOT-SIGMA
ul. Bartycka 20
00-950 Warszawa skrytka 1004

☐ Instytucje

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

NAZWA ZAKŁADU I ADRES POCZTOWY DO WYSYŁKI

Należność będziemy regulować z naszego konta

po otrzymaniu zamówionych numerów **mikroklanu**

PIECZĘĆ I PODPIS GŁÓWNEGO KSIĘGOWEGO

Istnieje już wiele pism o komputerach, ale jeśli chcesz wiedzieć więcej, jeśli interesują Cię komputery klasy PC, nowe programy, opisy zastosowań **KUP mikroklan**!

Jeśli **chcesz zobaczyć** na zdjęciach urządzenia o których czytasz - **KUP mikroklan**!

Jeśli chcesz mieć **czytelny** schemat urządzeń komputerowych **KUP mikroklan**!

Wysoka jakość treści i formy to **mikroklan**

PAMIĘTAJ!
Na **mikroklan** nie ma prenumeraty.
Zamówienie jest dogodnym i pewnym sposobem zakupu Twojego pisma

przewodów tworzących układ współrzędnych x-y. Zbliżenie urządzenia wskazującego do powierzchni deski powoduje wysyłanie sygnału do anteny. Antena umożliwia zamianę sygnału na współrzędne punktu (współpraca z odpowiednim programem) i precyzyjne określanie położenia wskaźnika.

Wśród innych metod określania położenia wskaźnika należy wymienić rozwiązanie, w którym analizowany jest sposób rozchodzenia się fali dźwiękowej. Metoda impulsów dźwiękowych nie wymaga płaszczyzny z anteną. Urządzenie wskazujące wysyła impulsy ultradźwiękowe, które są odbierane za pomocą mikrofonów (rys. 3). Położenie urządzenia wskazującego określane jest przez mikrofony na podstawie natężenia dźwięków. W odróżnieniu od innych rozwiązań w metodzie tej obszar, po którym można przemieszczać urządzenie wskazujące nie jest ściśle ograniczony.

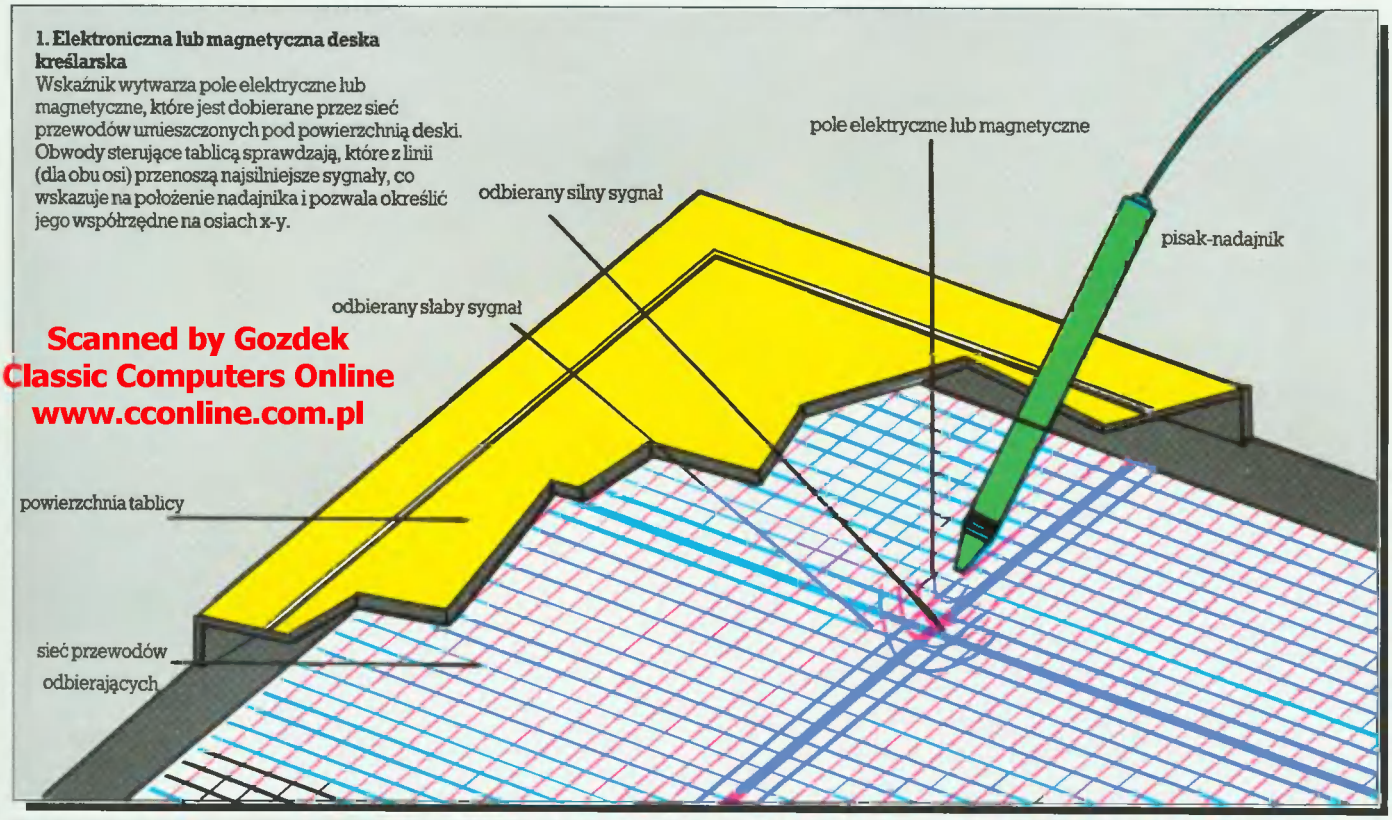
Inny sposób, w którym wykorzystywane są zmiany rezystancji, może zostać zrealizowany za pomocą układu dźwigni mechanicznych przenoszących ruch urządzenia wskazującego na potencjometry. Do metod rezystancyjnych należy też zaliczyć sposób, w którym wykorzystuje się czułą na nacisk poduszkę (rys. 2). Składa się ona z dwóch przewodzących warstw, które są zwierane w punkcie nacisku.

W zależności od prowadzonych prac dobierane są deski o różnych rozmiarach przestrzeni roboczej. Stosuje się też, zależnie od potrzeb, kilka metod transmisji współrzędnych x-y. Praca w trybie wektorowym umożliwia kreślenie odcinka przez wskazanie punktów początku i końca. Odpowiada to wektorowej metodzie opisu rysunków stosowanej w większości systemów CAD. W przypadku pracy w trybie ciągłym (np. przy wykonywaniu odręcznych szkiców lub kreśleniu krzywych) wysyłane są nieprzerwane sygnały odpowiadające zmieniającej się pozycji wskaźnika.

Na ogół tablice kreślarskie przyłączane są do wejścia szeregowego (ang. serial port) systemu PC, ale niektóre używają wejścia gier (ang. game port). Jedynie deska Pencept Penpad wymaga stosowania specjalnej płyty interfejsu. Informacje na temat używanych wejść i oprogramowania zostały zestawione w tabeli.

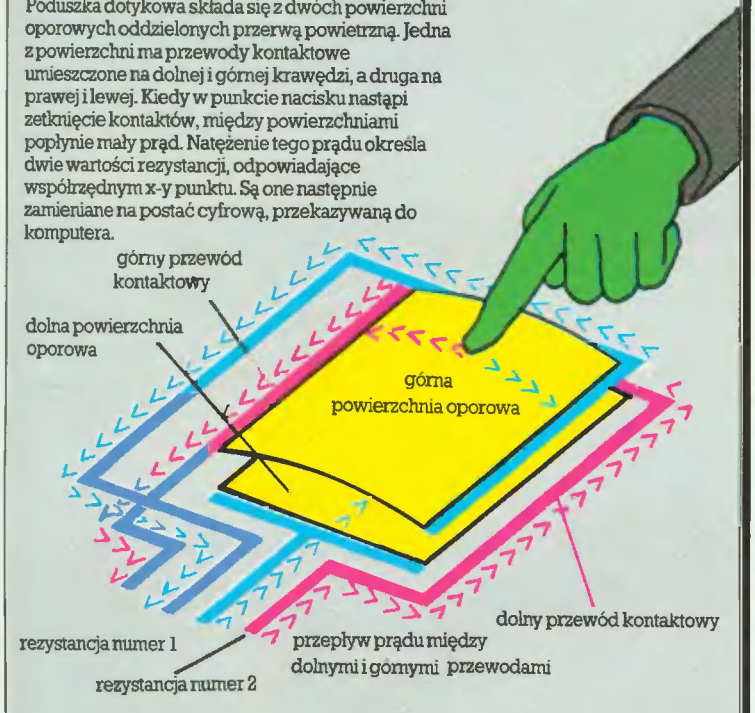
opr. Maciej Iwaszkiewicz wg. PC Magazine

1. Elektroniczna lub magnetyczna deska kreślarska
Wskaźnik wytwarza pole elektryczne lub magnetyczne, które jest dobierane przez sieć przewodów umieszczonych pod powierzchnią deski. Obwody sterujące tablicą sprawdzają, które z linii (dla obu osi) przenoszą najsilniejsze sygnały, co wskazuje na położenie nadajnika i pozwala określić jego współrzędne na osiach x-y.

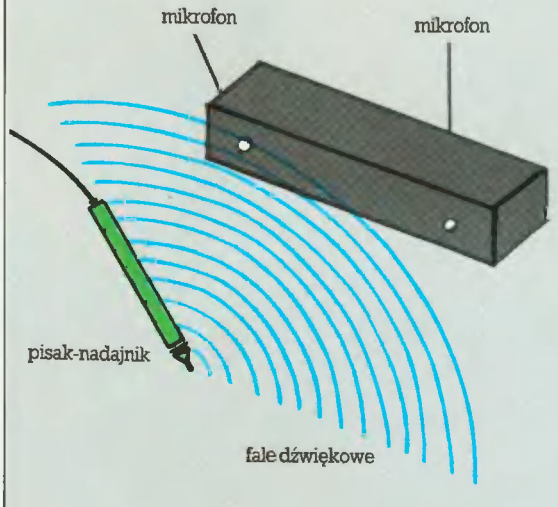


Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl

2. Rezystancyjna poduszka dotykowa.
Poduszka dotykowa składa się z dwóch powierzchni oporowych oddzielonych przerwą powietrzną. Jedna z powierzchni ma przewody kontaktowe umieszczone na dolnej i górnej krawędzi, a druga na prawej i lewej. Kiedy w punkcie nacisku nastąpi zetknięcie kontaktów, między powierzchniami popłynie mały prąd. Natężenie tego prądu określa dwie wartości rezystancji, odpowiadające współrzędnym x-y punktu. Są one następnie zamieniane na postać cyfrową, przekazywaną do komputera.



3. Akustyczna deska kreślarska
Pisak wysyła sygnał ultradźwiękowy, który jest odbierany przez mikrofony umieszczane w rogach obszaru roboczego. Mierzona jest siła sygnału odbieranego przez każdy z mikrofonów i na tej podstawie jest określana odległość pisaka od mikrofonu. Porównanie wartości sygnałów dla obu mikrofonów pozwala ustalić pozycję pisaka.



ISTOTNY JEST DOBÓR SPRZĘTU ...

Scanned by Gozdek
Classic Computers Online
www.cconline.com.pl